

Analýza přechodů pro chodce pomocí obrazů

Crosswalk Analysis Using Images

Bc. Marcel Jurišta

Diplomová práce

Vedoucí práce: Ing. Radovan Fusek, Ph.D.

Ostrava, 2021

Abstrakt

Cieľom tejto diplomovej práce je zistenie možností spracovania obrazu za účelom detekcie prechodov pre chodcov a porovnanie možných riešení. Táto diplomová práca je zameraná na spracovanie prechodov pre chodcov z pohľadu samoriaditeľných vozidiel. Pre vytvorenie funkčného detektoru je vhodné experimentálne otestovať viaceré prístupy spracovania obrazu. Hlavnými rozdielmi v navrhovaných detektoroch sú v predspracovaní dát. V prístupe s tréновaním je potrebné využiť dáta, ktoré však najkôr musia byť extrahované z dostupných materiálov. V súvislosti so spracovaním prechodu pre chodcov pomocou obrazov je v tejto diplomovej práci vykonané porovnanie existujúcich riešení, zistenie možností spracovania obrazu a navrhnutie spolu s implementáciou detektoru za pomoci zistených knižníc. Hlavnými časťami pre vytvorenie detektoru je extrahovanie dát pre tréновanie, ich predspracovanie a následné použitie vo vytvorení finálneho detektoru. Detektor zameraný na bezpečnosť chodcov je dôležitý v oblasti autonómneho riadenia vozidiel.

Klíčové slová

OpenCV, Tensorflow, Neurónové siete, Spracovanie obrazu, Mask-RCNN, Detekcia prechodov pre chodcov, Autonómne vozidlá

Abstract

The aim of this diploma thesis is to find out the possibilities of image processing for the purpose of crosswalk detection in order to detect pedestrian crossings and to compare possible solutions. This diploma thesis is focused on processing pedestrian crossings from the perspective of autonomous vehicles. To create a functional detector, it is advisable to experimentally test several approaches of image processing. The main differences in the proposed detectors are in data preprocessing. In a training approach, it is necessary to use data that must be extracted from available materials. In connection with the processing of pedestrian crossings using images, this thesis compares existing solutions, identifying the possibilities of image processing and design along with the implementation of the detector using the found libraries. The main parts for creating a detector are extracting the data for training, data pre-processing and subsequent use in creating the final detector. A pedestrian safety detector is important in the autonomous driving industry.

Keywords

OpenCV, Tensorflow, Neural networks, Image processing, Mask-RCNN, Crosswalk detection, Autonomous vehicles

Podakovanie

Veľmi rád by som podakoval vedúcemu diplomovej práce **Ing. Radovanovi Fusekovi, Ph.D** za odbornú pomoc, poskytnutie dát potrebných pre vytvorenie detektoru a konzultácie pri vytváraní tejto práce.

Obsah

Zoznam použitých symbolov a skratiek	6
Zoznam obrázkov	7
Zoznam tabuliek	9
1 Úvod	11
1.1 Motivácia	13
1.2 Autonómne vozidlá a detekcia prechodov	15
2 Prechody pre chodcov	19
3 Existujúce riešenia	21
3.1 Metódy spracovania obrazu bez trénovania	21
3.2 Metódy spracovania obrazu s trénovaním	25
3.3 Metódy detekcie prechodov bez spracovania obrazu	28
3.4 Využitie detekcie chodcov vzhľadom k lokalizácii prechodu	30
3.5 Porovnanie použitia rôznych prístupov	32
4 Vlastné riešenie	34
4.1 Návrh algoritmu bez trénovania a tvorba datasetu	34
4.2 Návrh algoritmu pre natrénovanie modelu pomocou neurónových sietí	40
5 Testovanie	46
5.1 Použitie modelu natrénovaného neurónovou sieťou	46
5.2 Použitie datasetov extrahovaných zo služieb tretích strán	48
5.3 Použitie spojenia neurónovej siete a prístupu bez trénovania	52
6 Možnosti zlepšenia	54
7 Záver	56

Literatúra	57
Prílohy	60
A Zoznam priložených súborov a zložiek a popis testovania kódu	61

Zoznam použitých skratiek a symbolov

BP	– Back Propagation
CHMI	– Context-based Movement History Image
CUDA	– Compute Unified Device Architecture
DCNN	– Deep Convolutional Neural Network
FC	– Fully-Connected
GPS	– Global Positioning System
GPU	– Graphics Processing Unit
HSV	– Hue Saturation Value
IDE	– Integrated Development Environment
IoU	– Intersect over Union
LMS	– Laser Measurement System
MCHOG	– Multiple-Cell Histogram of Oriented Gradient
MNIST	– Modified National Institute of Standards and Technology
OSM	– Open Street Maps
RAM	– Random Access Memory
RCNN	– Region Based Convolutional Neural Networks
RoI	– Region of Interest
SVM	– Support-vector machine
TP,TN,FP,FN	– True Positive, True Negative, False Positive, False Negative
USD	– United States Dollar
VB-DAS	– Vision-Based Driver Assistance Systems
XML	– eXtensible Markup Language

Zoznam obrázkov

1.1	Bezpečnosť zrakovo postihnutého chodca na prechode pre chodcov (vlastný zdroj)	14
1.2	Multisenzorový návrh podľa [24]	16
1.3	Znamenie autonómneho vozidla chodcovi [33]	17
1.4	Graf rýchlosti vozidla vo vzťahu k vzdialenosti od prechodu [32]	18
2.1	Typy prechodov podľa [4]	19
2.2	Model prechodu podľa [39]	20
2.3	Atypické druhy prechodov	20
3.1	Vzťah (<i>width</i> , <i>y</i>) jedného obrázku prechodu podľa [11]	22
3.2	Zobrazenie segmentácie obrazu podľa [11]	23
3.3	Postup detekcie podľa [18]	24
3.4	Výstup použitím neurónovej siete s vyznačením reálnych hodnôt [4]	25
3.5	Postup detekcie z OSM [26]	26
3.6	Spracovanie vstupného obrázku z OSM [26]	27
3.7	Označenie regresnej priamky (červená) v prípade bodov nachádzajúcich sa v prechode (zelená) a bodov nachádzajúcich sa na jednej čiare (žltá) [28]	28
3.8	(a) normálna skupina, (b) skupina označené ako jedno čiara, (c) skupina označená ako prechod pre chodcov [28]	28
3.9	Výstup z laseru [20]	29
3.10	Obdĺžnikový priebeh (modrá) a vstupný signál (červená) [20]	30
3.11	Diagram nutnosti uvedomenia si situácie podľa [36]	30
3.12	CMHI pre vyznačený pohyb chodcov podľa [36]	31
3.13	Ťažko identifikovateľné prechody	32
4.1	Ukážka hľadania HSV prahovania - obrázok pred a po aplikácií prahovania	35
4.2	Diagram postupu detekcie pomocou OpenCV	37
4.3	Ukážka výstupu algoritmu pre vytvorenie datasetu	38
4.4	Príklad anotácie pomocou služby imglab.in	39

4.5	Ukážka anotovaných dát v službe Imglab	40
4.6	Používateľské prostredie Google Colab	42
4.7	Hlavná architektúra siete Faster R-CNN	43
4.8	Vertikálne dopravné značky umiestnené pri prechodoch pre chodcov	45
5.1	Výsledky testovania datasetu 1. (zelenou čiarou sú označené anotované dáta, červenou výsledok detekcie)	48
5.2	Výsledky testovania datasetu 2. (zelenou čiarou sú označené anotované dáta, červenou výsledok detekcie)	48
5.3	Nový testovací dataset vytvorený pomocou služieb tretích strán	49
5.4	Správne vyhodnotené výsledky použitím neurónovej siete	50
5.5	Nesprávne vyhodnotené výsledky použitím neurónovej siete	50
5.6	Zle viditeľné alebo neúplné prechody a ich vyhodnotenie natrénovaným modelom . .	51
5.7	Správne alebo čiastočne správne vyhodnotené výsledky natrénovaným modelom . . .	51
5.8	FN výsledky pomocou natrénovaného modelu	52
5.9	Zobrazenie niektorých výsledkov testovania oboch prístupov (zelenou čiarou sú vý- sledky prístupu s tréningom, červenou sú výsledky algoritmu bez tréningu)	53
6.1	Zobrazenie dát vyhodnotených modelom natrénovaným pomocou vertikálnych do- pravných značiek prechodov (zelenou farbou model označuje pozdĺžne značenie, čer- venou vertikálne)	55
6.2	Zobrazenie dát vyhodnotených modelom natrénovaným pomocou vertikálnych do- pravných značiek prechodov 2 (zelenou farbou model označuje pozdĺžne značenie, červenou vertikálne)	55

Zoznam tabuliek

4.1	Porovnanie extrakcie dát z datasetov	38
5.1	Päť najlepších výsledkov testovania vlastných datasetov	47
5.2	Päť najhorších výsledkov testovania vlastných datasetov	47
5.3	Celkový priemer výsledkov testovacích dát	48
5.4	Testovanie pomocou algoritmu bez trénovania na datasete zo služieb tretích strán . .	50
5.5	Rozloženie správnosti testovacích dát pomocou oboch prístupov	52

Zoznam výpisov zdrojového kódu

3.1	Algoritmus pre spájanie segmentov čiar podľa [18]	24
4.1	Ukážka nastavenia a aplikácie prahovania v načítanom obrázku	36
4.2	Algoritmus pre načítanie ohraničujúcich obdĺžnikov z XML súboru	44

Kapitola 1

Úvod

Detekcia obrazu je v dnešnej dobe veľmi používaným pojmom v rôznych oblastiach priemyslu a zdravia. Asi najčastejšie je možné tento pojem počuť v spojení s autonómnymi vozidlami v oblasti automobilového priemyslu čo je aj hlavnou oblasťou použitia v súvislosti s touto diplomovou prácou. Autonómne alebo čiastočne autonómne vozidlá sa začínajú používať v mnohých krajinách sveta a pomaly si získavajú svoje miesto aj na našich cestách. S veľkým pokrokom však prichádza aj veľká zodpovednosť v oblasti bezpečnosti cestnej premávky, riešenia krízových situácií a minimalizácií rizík spojených s chybným vyhodnotením situácie. Počas jazdy sú v takmer reálnom čase vyhodnocované mnohé faktory. V obrazoch z rozličných kamier sú detekované rôzne objekty ako sú napr. ľudia, čiary na ceste, rôzne prekážky alebo napr. horizontálne značky na vozovke akými sú aj prechody pre chodcov. Ich rozpoznanie má v kombinácii s detekciou chodcov viesť k včasnému prispôbeniu rýchlosti vozidla, prípadne jeho úplnému zastaveniu. V spojení s detekciou chodcov je taktiež dôležité aby chodci, ktorí chcú prejsť cez prechod, mali istotu, že autonómne vozidlo ich zaregistrovalo a umožní im bezpečný prechod cez cestu.

Ďalšou možnosťou využitia implementácie detektoru je vyhodnocovanie obrazov z pohľadu chodcov. Táto téma môže mať veľký prínos najmä v oblasti zdravotníctva u zrakovo postihnutých ľudí. Vo veľkých mestách bývajú prechody cez veľké križovatky dobre prispôbené pre ľudí so zrakovým postihnutím formou zvukovej signalizácie. Menšie prechody na okrajoch menších miest alebo na odľahlých cestách, ktoré však nie sú vybavené elektronickým semafórom a signalizáciou pre chodcov môžu pre týchto ľudí predstavovať značné riziko. Aplikácia ktorá spracováva a vyhodnocuje signály z kamery môže výrazne zjednodušiť život najmä ľuďom, ktorých zrková porucha postihla počas ich života a musia sa tak prispôbiť novým nástrahám. V týchto situáciách by mohlo využitie moderných technológií značne prispieť k zníženiu rizika zdravotnej ujmy slabo vidiaceho alebo úplne nevidiaceho človeka a poskytnúť väčšiu istotu pri vykonávaní bežných činností ako je napr. chôdza po cestných komunikáciách.

V tejto diplomovej práci bude primárne spracovaný detektor určený pre samoriaditeľné vozidlá, avšak prístup akým je obraz vyhodnocovaný je veľmi podobný aj pre použitie zo strany chodcov. Pre

vytvorenie detektoru pre zrakovo postihnutých chodcov by sa javila ako vhodná možnosť použitia mobilného telefónu, prípadne tzv. smart doplnkov akými sú napr. okuliare. Pre spracovanie tejto diplomovej práce je však použitý počítač, ktorý disponuje oveľa väčším výpočtovým výkonom ako má mobilný telefón. Implementácia pre mobilné zariadenia by preto musela byť prispôbena tak, aby dokázala efektívne využívať jeho zdroje.

V rámci tejto diplomovej práce sú v kapitole 2 bližšie popísané druhy prechodov a ich najbežnejšie vyobrazenie na cestných komunikáciách. V kapitole 3 sú popísané projekty, ktoré sa zaoberajú problematikou spracovania prechodov pre chodcov alebo oblasti s tým spojené. V rámci kapitol 4 a 5 sú vytvorené a otestované dva druhy detektorov. Prvý využíva prístup bez trénovania a s druhým je ukázaná možnosť použitia neurónových sietí na detekciu prechodov. Výsledky sú zdokumentované a otestované a v kapitole 6 sú navrhnuté dodatočné kroky, ktoré by mohli viesť s zlepšeniu a spresneniu detekcie.

1.1 Motivácia

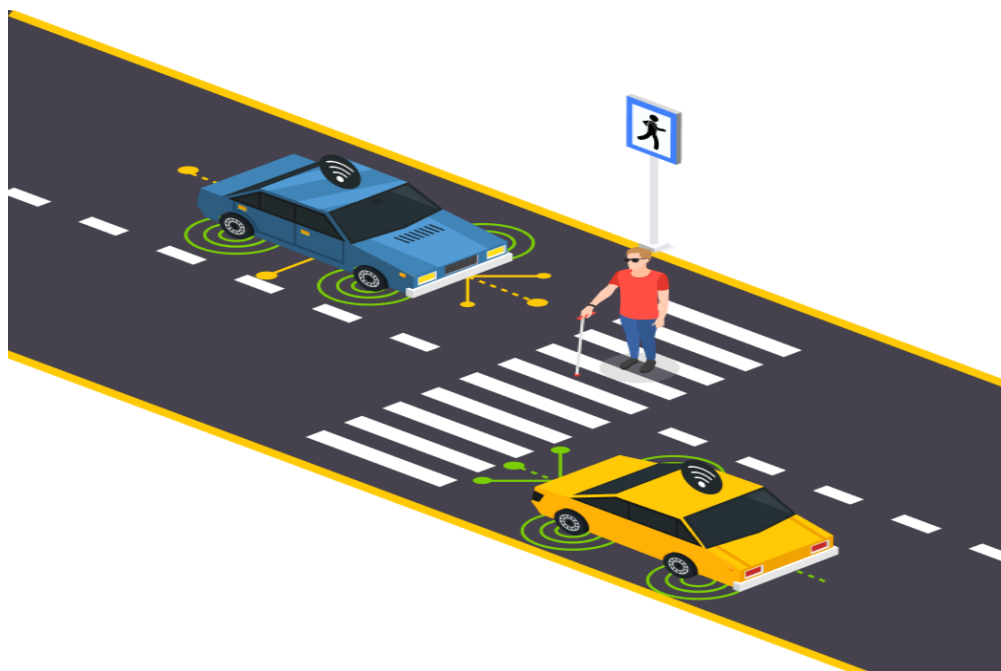
Hlavnú motiváciu pre skúmanie možností detekcie prechodu pre chodcov je možné hľadať v dvoch kľúčových oblastiach z čoho jedná z týchto oblastí je aj predmetom tejto diplomovej práce. Táto diplomová práca je zameraná výhradne na detekciu prechodov pre chodcov z pohľadu oblasti automobilového priemyslu a keďže v tejto oblasti hrá dôležitú úlohu z hľadiska bezpečnosti, je potrebné túto problematiku preskúmať samostatne. Podľa [41] sú križovatky, prechody a miesta, v ktorých sa cesty rozdeľujú veľmi bežnou, no pre oblasť samoriaditeľných áut aj značne dôležitou súčasťou cestných komunikácií. Túto informáciu podporuje aj fakt, že miera výskytu dopravných nehôd je v týchto oblastiach veľmi vysoká. Podľa [34] každoročne zomrie pri dopravných nehodách viac ako 1.2 milióna ľudí. Tieto nehody sú z 96% spôsobené ľudským faktorom (ako napr. nepozornosť vodiča). Mení sa taktiež preťaženie našich miest, keďže viac ako 20% komunikácií je zvyčajne určených pre parkovanie. V posledných rokoch sa taktiež urobil obrovský pokrok v oblasti autonómnych vozidiel v dôsledku veľkých prielomov v umelej inteligencii a v hardvéri. Pri autonómnej jazde vozidla môže deliť správne vyhodnotenie rizikovej situácie medzi životom a smrťou. Korektnosť vyhodnotenia situácie však stojí na mnohých faktoroch a výsledkoch no rozpoznanie prechodu pre chodcov a následne aj chodcov samotných je dôležitým ukazovateľom pre správne vyhodnotenie situácie. Využitie v oblasti automobilového priemyslu však neostáva iba v oblasti samoriaditeľných áut. Spracovanie obrazov počas jazdy je dôležité aj v asistenčných systémoch vozidla určených pre včasné upozornenie vodiča o možných rizikách nachádzajúcich sa na vozovke alebo mimo nej, prípadne na čiastočné alebo úplné obmedzenie jazdy zo strany asistenčného systému v prípade vyhodnotenie krízovej situácie. Takéto systémy môžu v značnej miere prispieť k bezpečnosti všetkých účastníkov cestnej premávky a znížiť riziká spojené s prítomnosťou ľudského faktoru.

Nie menej zaujímavou a taktiež veľmi dôležitou oblasťou záujmu by bolo využitie projektu pre zrakovo postihnutých ľudí, kde by bolo potrebné sa pozeráť na ich problémy z viacerých rozličných aspektov. Podľa [9] rozlišujeme niekoľko rozličných techník, ktoré sa používajú pre zaistenie bezpečnosti nevidiacich ľudí:

- slepecká palica
- detekovateľné varovné materiály
- varovné zvukové signály
- rečové značky
- aplikácia skúmajúca GPS pozíciu

Na obrázku 1.1 je možné vidieť, ako by mohla vyzeráť pomoc zrakovo postihnutým ľuďom v praxi. Nakoľko držanie smartfónu v rukách zrakovo postihnutého nemusí byť príliš bezpečné a

efektívne, mohlo by byť podobné zariadenie implementované v rámci slepeckej palice, prípadne sa naskytuje možnosť využitia inteligentných okuliarov.



Obr. 1.1: Bezpečnosť zrakovo postihnutého chodca na prechode pre chodcov (vlastný zdroj)

Práve využitie inteligentných okuliarov je popísané v článku [13] kde sa označuje ako potenciálna možnosť využitia v oblasti detekcie prostredia pre ľudí trpiacimi rôznymi poruchami. Väčšina zrakovo postihnutých chodcov má podľa [11] problém prekonať križovatky využitím bežných orientačných a pohybových schopností. Môže byť pre nich veľkým problémom určiť správny smer prechodu a zosynchronizovať tak svoju chôdzu práve so smerom prechodu a zamedziť tak riziku vyjdenia chodca mimo vyznačeného prechodu. Práve s týmto by mohlo výrazne pomôcť rýchle spracovanie obrazu pomocou dostupných technológií, akými sú napr. smartfóny alebo energeticky nenáročné jedno alebo viac účelové zariadenia vybavené kamerami. Pre lepšie pochopenie problematiky detekcie prechodu pre chodcov či už z pohľadu zrakovo postihnutých ľudí alebo z oblasti automobilového priemyslu je potrebné zistiť, aké sú dostupné existujúce riešenia, aké sú spôsoby analýzy a detekcie obrazu a aké možnosti sa ponúkajú v rôznych metódach detekcie prechodov.

V rámci tejto diplomovej práce budú vytvorené algoritmy, ktoré predchádzajú alebo priamo súvisia s detekciou alebo testovaním detektoru prechodov. Hlavnou motiváciou pre vytvorenie detektoru, či už zo strany autonómnych vozidiel alebo z pohľadu zrakovo postihnutých chodcov je bezpečnosť na cestných komunikáciách. V oboch prípadoch včasná detekcia znižuje riziko kolízie chodca s vozidlom. Pre túto diplomovú prácu bol vybraný detektor umožňujúci detekovať prechod primárne z pohľadu autonómnych vozidiel. Doplnením datasetu o snímky z pohľadu chodcov by bolo možné prispôbiť detektor na detekciu prechodov pre zrakovo postihnutých chodcov.

1.2 Autonómne vozidlá a detekcia prechodov

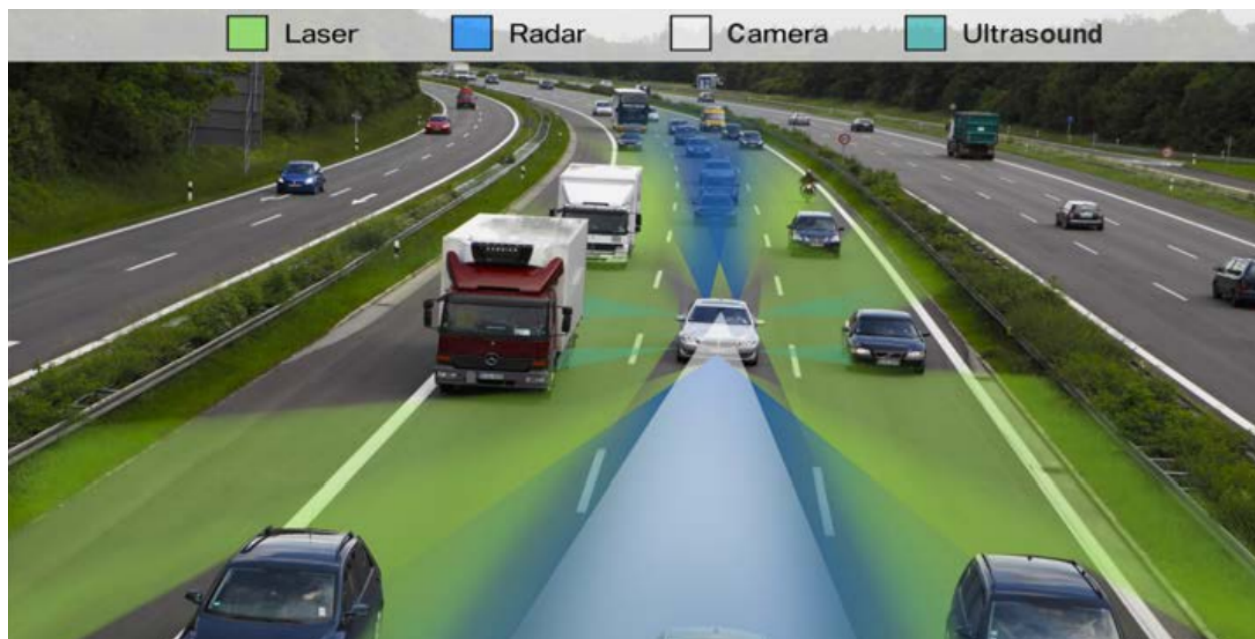
S vývojom nových technológií a so stúpajúcou popularitou autonómnych vozidiel je nutné začať myslieť na integráciu autonómneho systému do bežnej premávky [30]. HS Automotive Inc. predpokladá že predajnosť autonómnych vozidiel globálne vzrastie až na hodnotu 21 miliónov do roku 2035. Zároveň sa podľa [32] trh s autonómnymi vozidlami dosiahne do roku 2035 na hodnotu 77 miliárd USD. Takmer všetci výrobcovia áut investujú nemalé finančné prostriedky na výskum autonómnych vozidiel. Taktiež sa podľa [5] predpokladá, že autonómne vozidlá by mohli eliminovať 90% dopravných nehôd v Spojených Štátoch Amerických a ušetriť až 190 miliárd USD na poplatkoch spojených s poškodením majetku a zdravia a zachrániť tisíce životov. V cestnej premávke vznikajú interakcie medzi vodičmi, chodcami a cyklistami a bežne zaužívané pravidlá a znamenia je potrebné previesť aj do oblasti autonómnych vozidiel. Podľa [33] sa však chovanie chodcov alebo iných účastníkov cestnej premávky mení vzhľadom k demografii chodcov, dynamike premávky, prostrediu a pod. Interakcia medzi účastníkmi cestnej premávky hrá dôležitú rolu v jej fungovaní a v jej plynulosti. Ak sa napr. v rušných hodinách na križovatke z vedľajšej cesty nachádza veľké množstvo áut, ktoré majú problém sa zaradiť, častokrát sa stane, že sa nájde ohľaduplný vodič a dá znamenie vodičovi čakajúcemu na vedľajšej ceste prebliknutím svetiel alebo jednoduchým gestom. Týmto dáva druhému vodičovi najavo, že o ňom vie a že mu umožní bezpečné zaradenie do premávky. Podobné situácie môžu nastať aj pri detekcii prechodu a chodcov nachádzajúcich sa blízko neho. Autonómne vozidlo musí spĺňať základné kritéria pre zachovanie bezpečnosti a plynulosti cestnej premávky.

Podľa [10] je možné autonómne vozidlá klasifikovať do šiestich rôznych levelov:

- Level 0 - Bez automatizácie: vodič má plnú kontrolu nad vozidlom a reaguje na automatizované systémy vo vozidle
- Level 1 - Asistencia vodičovi: automatizovaný systém aj vodič zdieľajú zodpovednosť pri riadení. Od vodiča sa očakáva prevzatie plnej kontroly nad vozidlom v akomkoľvek momente
- Level 2 - Čiastočná automatizácia: automatizovaný systém prevezme plnú kontrolu nad vozidlom. Vodič však musí počas celej jazdy monitorovať stav a byť pripravený okamžite prevziať kontrolu
- Level 3 - Podmienená automatizácia: Vodič sa môže vzdať niektorých činností počas jazdy a systém informuje vodiča o prevzatí kontroly nad týmito činnosťami v obmedzenom čase
- Level 4 - Vysoká automatizácia: vozidlo vykonáva všetky činnosti samostatne a vodičova pozornosť nie je vyžadovaná. Ľudská asistencia je umožnená len za špecifických podmienok
- Level 5 - Plná automatizácia: Autonómny systém je identický s ľudským vodičom v akomkoľvek vodičskom úkone. Počas jazdy nie je vyžadovaná žiadna asistencia vodiča.

V tejto diplomovej práci teda pojem autonómne vozidlo spadá pod kategóriu 4 až 5 zoznamu 1.2.

Dnešné autonómne vozidlá používajú tzv. Vision-Based Driver Assistance Systems (skrátene VB-DAS) ktoré patria do asistenčných systémov vozidla. Podľa [24] VB-DAS kombinujú signály z viacerých kamier a senzorov. Na obrázku 1.2 je možné vidieť návrh multisenzorového snímania vozidla.



Obr. 1.2: Multisenzorový návrh podľa [24]

Pre lepšie pochopenie potreby detekcie prechodov bude uvažovaná vzorová situácia, kedy sa vozidlo pohybuje mimo obce na úseku, kde sa nachádza prechod pre chodcov a rýchlosť premávky je obmedzená na 70 km/h. Pred prechodom sa nachádza vegetácia, ktorá zabraňuje včas identifikovať možných chodcov, ktorí budú chcieť prechod použiť. Kvôli pomerne vysokej maximálnej povolenej rýchlosti by mohlo mať náhle zastavenie autonómneho vozidla po identifikácii ľudí nachádzajúcich sa blízko prechodu katastrofálne následky. Chodci nachádzajúci sa v blízkosti prechodu by boli systémom zaregistrovaní avšak pri podobných úsekoch mimo obce kde je navyše rýchlosť obmedzená z pôvodných 90 km/h častokrát vodiči nedodržiavajú bezpečný odstup a reakčná doba pri tak prudkom zastavení nemusí byť dostatočne rýchla. Je preto dôležité, aby systém na detekciu prechodov spolupracoval s inými systémami a ich dáta boli spracované súbežne. V našom prípade by mala byť braná v úvahu napr. aktuálna rýchlosť vozidla, informácie o jazde za vozidlom a o ostatných vozidlách a v neposlednom rade informácie o tom, či sa v blízkosti prechodu nachádzajú chodci, ktorí sa ho chystajú použiť.

Plynulosť a interakciu účastníkov cestnej premávky je možné demonštrovať na jednoduchšej situácii, do ktorej sa už určite dostal každý. V mestách sa nachádza mnoho prechodov a rýchlosť akou sa môže auto pohybovať sa častokrát nedostane ani na maximálne povolené limity. Vodič

bežného vozidla zväčša pred prechodom zastaví a už behom zastavovacieho manévru dá chodcom čakajúcim pred prechodom znamenie jednoduchým gestom alebo prebliknutím svetiel. Chodec má teda istotu že vodič o ňom vie a môže bezpečne použiť prechod. Aby bola zachovaná plynulosť a naučené nepísané pravidlá, systém autonómneho vozidla by mal vedieť vytvoriť interakciu s chodcami podobne, ako by to urobil reálny šofér. Takéto situácie sú pre ľudí normálne a podvedome ich vnímajú ako bežnú súčasť cestnej premávky avšak algoritmom, ktoré riadia chovanie vozidla je nutné tieto návyky implementovať. V štúdií [33] je rovnako popísaný aj experiment zameraný na správanie chodcov.

V experimente vykazoval šofér rôzne stavy pozornosti a zisťovala sa tak reakcia chodcov ktorí chceli použiť prechod. Pokiaľ šofér venoval pozornosť riadeniu, všetci chodci sa rozhodli prejsť cez prechod. V prípade, kedy sa šofér venoval svojmu smartfónu 20% chodcov neprešlo cez prechod a v prípade, kedy vodič čítal noviny alebo sa v aute vôbec nenachádzal, až 60% chodcov prechod nepoužilo. V tomto experimente sa teda ukázalo, že pri očnom kontakte cítia chodci najväčší pocit bezpečia a naopak, v prípade že sa v autonómnom vozidle vodič nenachádza, je potreba ich dôveru zabezpečiť napr. spôsobom zobrazeným na obrázku 1.3.



Obr. 1.3: Znamenie autonómneho vozidla chodcovi [33]

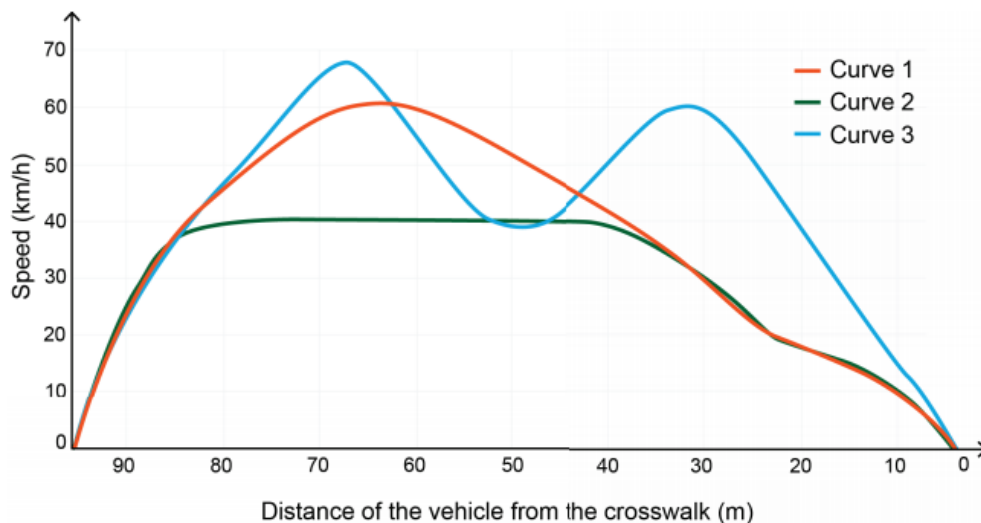
Iný experiment ktorý bol predstavený v [32] ukazuje využitie dnes veľmi populárnej virtuálnej reality pre vykonanie testov. Prieskum ukazuje že interakcia medzi chodcami a vodičmi, podobne ako bolo poznamenané v štúdií [33], vážne závisí na očnom kontakte. Iné signály, akými sú napríklad postoj a pohyb tela pomáhajú vodičovi pochopiť zámary chodca [35]. V štúdií [32] bola prihliadnuc na problémovú interakciu medzi vodičom a chodcom sformovaná nasledujúca otázka:

Ako dokáže neprítomnosť ľudského vodiča vo vozidle ovplyvniť interpretáciu chovania vozidla za rôznej viditeľnosti z pohľadu chodcov?

Práve tento problém je možné študovať použitím virtuálnej reality. Virtuálna realita je definovaná ako počítačovo generované prostredie kde používateľ môže zažiť prítomnosť virtuálneho sveta bez nutnosti byť v ňom reálne prítomný.

Výhodou testu pomocou virtuálnej reality je bezpečnosť chodcov a počítačová kontrola nad virtuálnymi vozidlami. Zároveň je potrebné uvedomiť si, že v teste pomocou virtuálnej reality nie je nutné mať v aute vodiča, zatiaľ čo pri teste v ostrej premávke by táto možnosť nebola reálna z dôvodu legislatívy ktorá v čase písania článku stále neumožňovala jazdu plne autonómneho vozidla po oficiálnych cestných komunikáciach.

V tomto experimente bolo zúčastnených 15 ľudí (10 mužov a 5 žien). 12 účastníkov bolo zo Švédska a 3 boli z Ázie. Experiment bol vykonaný s každým účastníkom samostatne. Odohrával sa v miestnosti 5x5m kde boli pomocou špeciálneho setu simulované tri rôzne prostredia (viď. obr. 1.4) za dvoch rozličných podmienok. Výsledkom je teda 6 nezávislých testov. Na obrázku 1.4 je možné vidieť rýchlosť vozidla v troch rôznych prostrediach vzhľadom k vzdialenosti vozidla od prechodu. Počas experimentu sa s chodcami komunikovalo a boli zaznamenávané ich nálady a pocity bezpečnosti. Zúčastneným nebolo zverené že sa jedná o autonómne vozidlo aby výsledky neboli skreslené alebo ovplyvnené.



Obr. 1.4: Graf rýchlosti vozidla vo vzťahu k vzdialenosti od prechodu [32]

Z výsledkov testu vyplýva že zúčastnení najhoršie reagovali na kolísavú rýchlosť v prípade 3. krivky obrázku 1.4. Veľkú rolu hralo aj počasie a dokonca aj národnosť. Na otázku, či si myslia že vozidlo bolo autonómne odpovedali v prípade 3. krivky negatívne. Oddôvodnené to bolo názorom, že autonómne vozidlo by nikdy takým spôsobom nejazdilo. Z tohoto výskumu teda vyplýva okrem iného aj fakt, že ľudia môžu autonómnym vozidlám dôverovať v prípade, že budú prispôbolené v oblasti komunikácie s chodcami.

Kapitola 2

Prechody pre chodcov

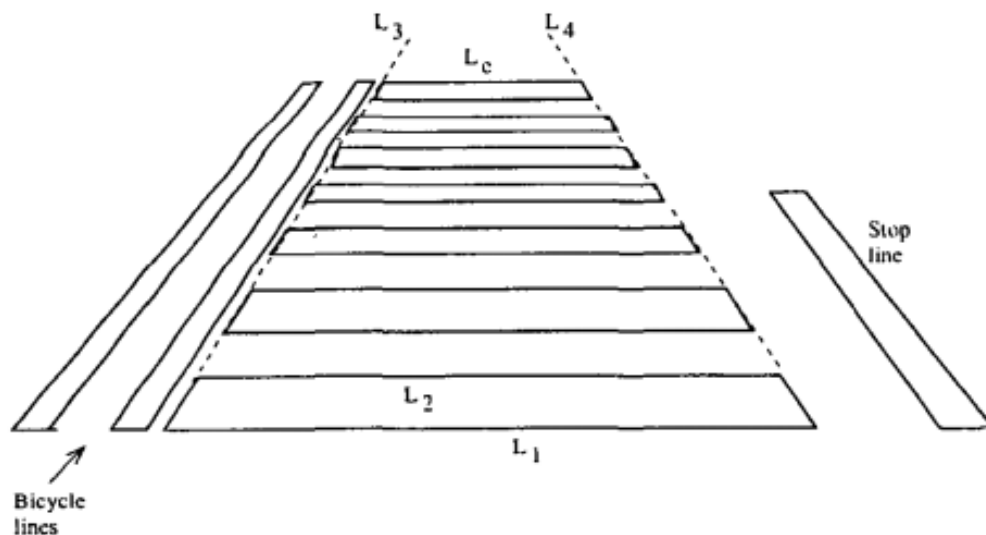
Vychádzajúc z [4] existuje viacero druhov prechodov, z čoho dva sú štandardne využívané. Pre porovnanie druhov prechodov (viď obrázok 2.1) na ktorom je zobrazený štandardný typ prechodu, prechod typu zebra a hybridný prechod spájajúci dva predošlé typy. Táto diplomová práca je zameraná na prechody typu zebra a v krajinách Česká a Slovenská republika sú spolu s hybridnými prechodmi najviac zastúpené práve tieto druhy prechodov. V tejto diplomovej práci teda pod pojmom štandardné typy prechodov rozumieme prechody typu zebra.



Obr. 2.1: Typy prechodov podľa [4].

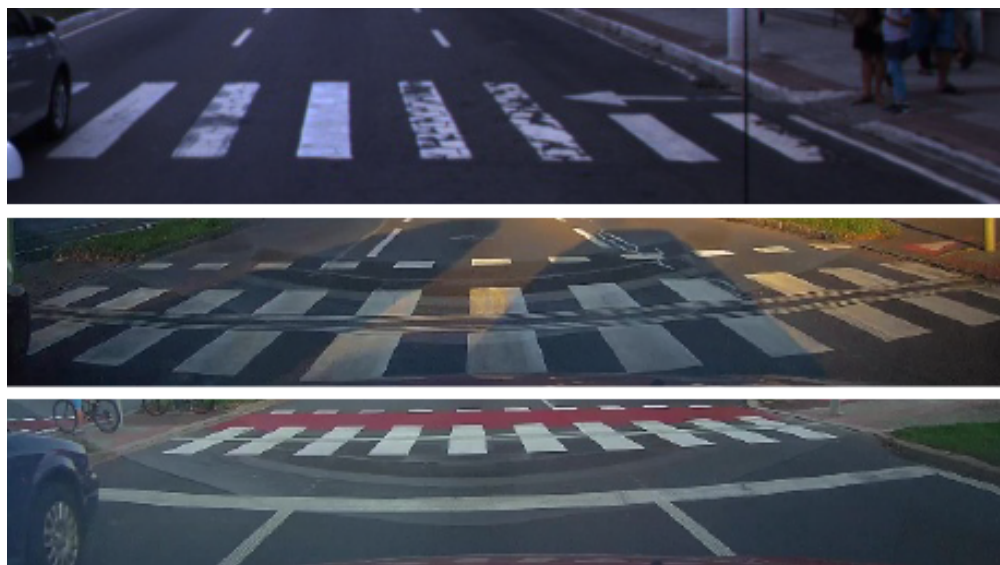
Práve štandardné prechody majú mnoho spoločných vlastností, ktoré môžu pomôcť s ich identifikáciou. Jedným z príznakov sú v prípade postačujúcej kvality obrázku a nízkej amortizácií prechodu dobre detekovateľné hrany. Problémy s amortizáciou však veľmi sťažujú detekciu pomocou bežných prístupov, ako bolo popísané v predošlých kapitolách. Ďalšou vlastnosťou prechodov je fakt, že pruhy prechodu sú typicky oddelené rovnakou vzdialenosťou vozovky. Línie prechodu sú v reálnom svete rovnobežné a rohy prechodov by mali tvoriť rovnú líniu, avšak pri spracovaní obrazu je potrebné počítat s nežiadúcimi deformáciami.

Pre jasný popis prechodu pre chodcov detekovaného v tejto diplomovej práci bude použitý obrázok popísaný autormi v práci [39]. Model prechodu je definovaný ako periodicky sa opakujúce biele (zriedkavo žlté) čiary ilustrované v obrázku 2.2.



Obr. 2.2: Model prechodu podľa [39].

L_1 určuje prvý pruh prechodu ktorý je najbližšie pozorovateľovi (v tomto prípade by šlo pravdepodobne o chodca), L_e je posledným pruhom prechodu. L_3 a L_4 sú bočnými hranicami línie prechodu. Zriedakedy sú v prechode vyznačené aj na prechod kolmé pruhy pre cyklistov alebo zastavovacie pruhy pre vozidlá.



Obr. 2.3: Atypické druhy prechodov.

Existujú však aj prechody, ktoré sa vymykajú bežnému popisu prechodu. Na obrázku 2.3 je možné vidieť niekoľko atypických prechodov použitých pri spracovaní datasetu tejto diplomovej práce.

Kapitola 3

Existujúce riešenia

Téma detekcie prechodu pre chodcov má najväčšie zastúpenie práve v dvoch spomínaných odvetviach. Pre získanie lepšieho prehľadu nad touto problematikou je potrebné odkazovať sa na dostupné práce zaoberajúce sa témami detekcie prechodov pre chodcov. Niekoľko organizácií alebo jedincov sa podobnou problematikou zaoberalo a výsledky sú zdokumentované v prislúchajúcich citáciach. V dostupných riešeniach bude možné vidieť rôzne prístupy k spracovaniu obrazov z viacerých rôznych pohľadov. Okrem už popísaných princípov v kategórii 1.1 pri ktorých sú využívané dáta z pohľadu chodcov, je možné vidieť aj spracovanie leteckých záberov z verejne prístupnej služby OpenStreetMap a využitie prístupov bez trénovania aj prístupov s trénovaním. V ďalších podsekciiach je popísané oboznámenie sa s problematikou detekcie prechodov a s možnými spôsobmi spracovania obrazu. V časti 3.1 je možné vidieť prehľad riešení bez potreby trénovania vytvoreného datasetu. Prístupy s trénovaním sú zhrnuté v časti 3.2 a v časti 3.3 sú popísané metódy, ktoré pre detekciu prechodov nepoužívajú spracovanie obrazu. V kapitole 3.4 je ukázané aj využitie detekcie chodcov vzhľadom k lokalizácii prechodu a rôzne prístupy sú v časti 3.5 porovnané. Ďalej sú popísané aj postupy použité v zdrojových kódach dostupných na verejných repozitároch a zároveň je analyzovaná použiteľnosť jednotlivých úkonov v navrhovanej aplikácii.

3.1 Metódy spracovania obrazu bez trénovania

Metódy zaoberajúce sa detekciou prechodov boli popísané v niekoľkých projektoch, ktorých prislúchajúce sprievodné texty sú bližšie popísané nižšie. Tieto metódy sa väčšinou vyznačujú vyššou rýchlosťou a nižšími požiadavkami na spracúvajúci hardvér, preto sú pomerne vhodné na spracovanie v takmer reálnom čase.

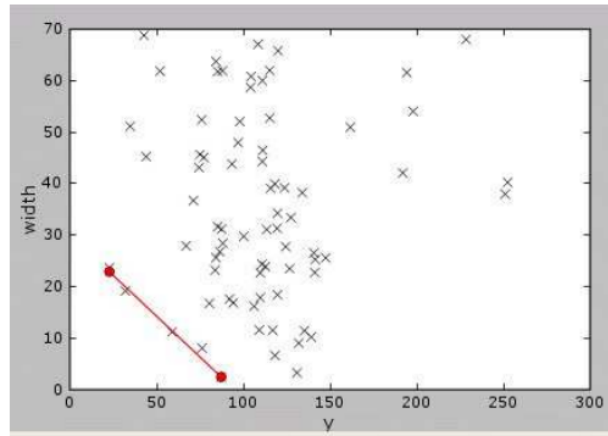
3.1.1 Houghova transformácia v kombinácii so segmentáciou obrazu

O detekcii prechodov bolo vydaných niekoľko článkov ako napr. [37] alebo [38], v ktorých sa typicky využíva Houghova transformácia pre extrakciu okrajov prechodu. Podľa [11] sú však podmienky

v reálnom svete o niečo komplikovanejšie. Preplnené scenérie s rôznymi tieňmi, saturačné výkyvy a zaoblenie čiar na prechode často krát spôsobujú, že Houghová transformácia je neadekvátne. Preto namiesto spoliehania sa na zhlukovanie globálnych štruktúr, ako to je v prípade Houghovej transformácie autori používajú vo svojom článku lokálne zoskupovanie založené na segmentácii obrazu. Tá bola v minulosti úspešne použitá na detekciu špecifických objektov a oblastí záujmov z pozadia. Štandardné techniky sú použité napr. na nájdenie tlačeneho textu, prúžkovaných vzorov, budov alebo vegetácie. Model ktorý využívajú autori je založený na segmentácii obrazu použitím geometrických vzťahov vlastností extrahovaných zo vstupných dát.

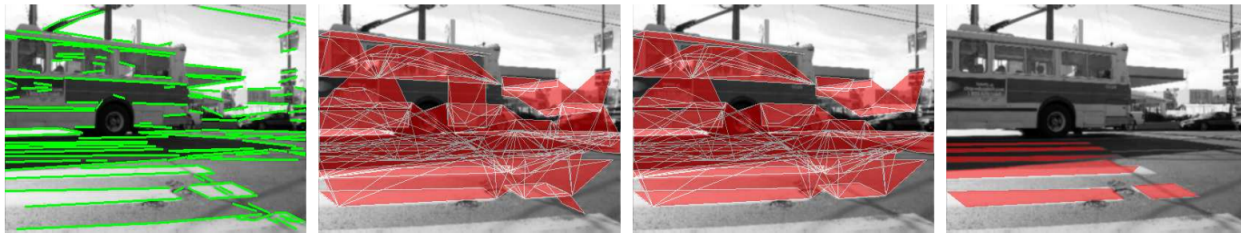
Každý element obrázku je klasifikovaný pomocou grafu ako objekt alebo zem. V tomto prístupe sa počíta s tým, že čiary prechodu majú podobné geometrické vlastnosti. Autori využívajú techniku Belief Propagation (BP) ktorá zabezpečí prenesenie informácie každému uzlu grafu. BP sa používa v grafických modeloch pre správne nastavenie marginálnych pravdepodobností napr. v stromových štruktúrach. Pre bližšie pochopenie BP a jej funkčnosti viď. [44]. Po extrakcii uzlov ktoré predstavujú pruhy prechodu sú použité unárne a binárne operátory pre vyhodnotenie vhodnosti označenia elementu ako objekt.

Pri extrakcii sa uvažuje fakt, že pri pohľade kamery sa šírka čiar prechodu mení so vzdialenosťou od snímajúcej kamery. Predpokladá sa teda že čiary, ktoré sú bližšie k fotografovi sú na obrázku nižšie. v prípade empirickej distribúcie všetkých pruhov (a nie iba tých, ktoré ležia na prechode), je možné zistiť že existuje vzťah medzi šírkou čiar *width* a vertikálnou súradnicou *y*. Na obrázku 3.1 je možné vidieť vzťah hodnôt *width*, *y* pruhov extrahovaných z jediného obrázku prechodu pre chodcov. Popísaný postup je taktiež s malými obmenami dostupný aj v [22], kde bol podobný algoritmus s využitím faktorových grafov úspešne implementovaný pre Noki N95.



Obr. 3.1: Vzťah (*width*, *y*) jedného obrázku prechodu podľa [11]

Na obrázku 3.2 je možné vidieť postupnú segmentáciu a extrakciu geometricky podobných vlastností ktoré prechody pre chodcov vykazujú. Jednotlivé kroky sú na obrázku 3.2 zobrazené zľava doprava:



Obr. 3.2: Zobrazenie segmentácie obrazu podľa [11]

- filtrovanie segmentov rovných čiar
- párovanie segmentov čiar tvoriacich štvruholníky
- uzly v grafickom modeli
- uzly označené ako figure po aplikovaní BP

Popísaný prístup podľa autorov funguje rýchlo (rádovo v jednotkách sekúnd na jeden obrázok) a javí sa ako vhodný pre použitie najmä v oblasti zdravotníctva a už spomínanej pomoci zrakovo postihnutým ľuďom.

3.1.2 Obdĺžnikove priznaky

Autori v článku [18] popisujú detekčnú stratégiu určenú pre asistenčné systémy vozidla. Táto metóda by mala zabezpečiť spoľahlivú detekciu s použitím kamery upevnenej na vozidle. Základnou myšlienkou detekcie podľa autorov je použitie adaptívnych Haarových filtrov, kde priepustnosť filtra je upravovaná podľa vzdialenosti objektu od vozidla. Výsledky sú následne odovzdané klasifikačnému modulu, ktorý vykoná klasifikáciu. Finálna detekcia je následne sledovaná štandardným Kalmanovým filtrom.

Prvým krokom systému detekcie je adaptívne filtrovanie obrázku použitím Haar-like features. Jej cieľom je zvýšiť viditeľnosť prechodu a redukovať ostatné objekty v obrázku. Ideálny prechod môže byť popísaný kombináciou obdĺžnikov s rovnakými rozstupmi. Vďaka tomu je možné zostrojiť lineárny filter, ktorý bude identický s frekvenčnou odozvou prechodu. Je však potrebné myslieť na úpravu filtra závislú na vzdialenosti od prechodu, keďže medzery medzi čiarami sa v obraze zväčšujú so znižujúcou sa vzdialenosťou vozidla od prechodu.

V práci autorov je použitý filter popísaný rovnicou

$$\bar{f}(x) = f(x) * h_k^l(x) \quad (3.1)$$

kde k označuje poradie derivačného filtra a l označuje poradie integrácie.

Po filtrácii obrázku musia byť detekované oblasti prechodu. V tomto kroku je použitá segmentácia. Vyfiltrovaný obrázok je najskôr prevedený na binárny obrázok. Keďže filtrovanie eliminovalo

takmer všetky oblasti, kde sa nenachádzajú čiary prechodu, nie je nutné vykonávať nájdenie kontúr. V algoritme 3.1 je možné vidieť postup, akým sú spájané segmenty čiar.

```

for all rows  $y$  do
     $\overline{bw}(x) \leftarrow bw(x) * \sum_{i=0}^{b-1} d(x - iW_{px})$ 
    for all columns  $x$  do
        if  $\overline{bw}(x) == b$  AND  $\overline{bw}(x + \frac{W_{px}}{2}) = 0$  then
            for  $i = x$  TO  $x + (b - 1)W_{px}$  do
                 $o(x, y) \leftarrow 1$ 
            end for
        end if
    end for
end for

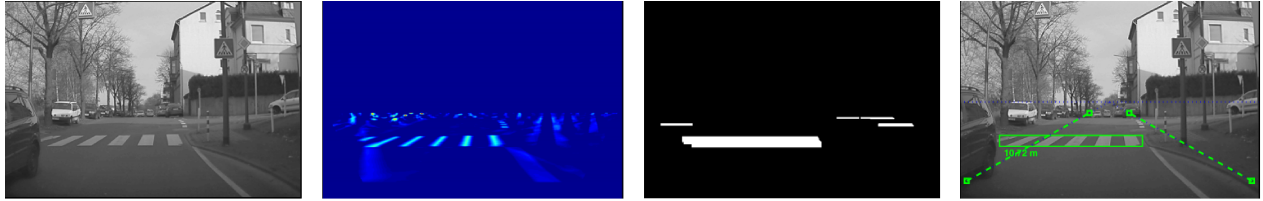
```

Výpis 3.1: Algoritmus pre spájanie segmentov čiar podľa [18]

Riadok binárneho obrázku je definovaný $bw(x)$. Binárny obrázok je konvolvovaný funkciou

$$\sum_{i=0}^{b-1} d(x - iW_{px}) \quad (3.2)$$

kde $d(x)$ je diskretnou impulznou funkciou. Hodnota b určuje minimálny počet susedných línií ktoré tvoria jeden región. Pre dosiahnutie najlepších výsledkov by hodnota b mala byť menšia ako 2 a väčšia ako počet segmentov bežného prechodu. Výsledok konvolúcie je označený ako $\overline{bw}(x)$. V algoritme sa ďalej vyskytuje $o(x, y) \leftarrow 0$, čo je výstupný binárny obrázok a W_{px} čo označuje medzeru medzi líniami v jednotkách px.



Obr. 3.3: Postup detekcie podľa [18]

Na obrázku 3.3 je možné vidieť jednotlivé kroky až k výslednej klasifikácii prechodu a vyznačeniu prechodu spolu so vzdialenosťou od prechodu a smerom vozovky. Po klasifikácii ďalej nasleduje finálny stav systému detekcie - sledovanie pomocou Kalmanovho filtra. Viac informácií o problematike Kalmanovho filtra je možné nájsť na [43].

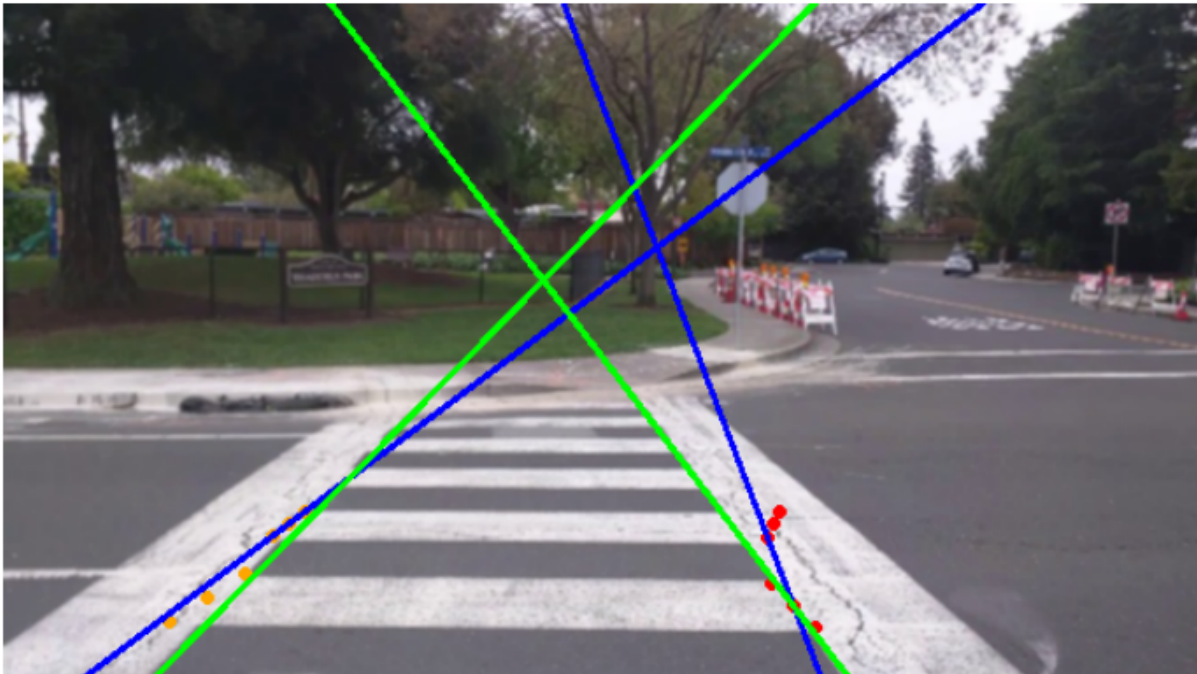
V tomto projekte bol úspešne aplikovaný systém detekcie prechodov ktorý zahŕňa adaptívne filtrovanie pomocou obdĺžnikových priznakov, segmentáciu obrazu pomocou algoritmu 3.1, klasifikáciu a následné sledovanie pomocou Kalmanovho filtra.

3.2 Metódy spracovania obrazu s trénovaním

Pre vyhodnotenie pozície prechodu pre chodcov v obraze sa naskytuje možnosť využitia prístupov s trénovaním. Overenie použiteľnosti v požadovanej oblasti je zdokumentované v nasledujúcich podsekcách.

3.2.1 Detekcia prechodov použitím neurónových sietí

V projekte ktorého sprievodný text je dostupný pod názvom "Zebra Crosswalk Detection Assisted By Neural Networks" [4] je možné vidieť detekciu prechodov za pomoci datasetu vytvoreného z reálnych dát a s následným natrénovaním neurónovej siete pomocou tohto datasetu. Dáta použité pre dataset boli podľa [4] výberom z videí natáčaných vo viacerých križovatkách a priesečníkoch ulíc. Videá boli vytvorené v rôznych častiach dňa od skorého rána až po neskorý večer. Videá boli zhotovené z rôznych uhlov a pozícií. Niektoré z videí boli označené pre použitie iba na účely tréovania a niektoré na účely evaluácie. V tomto projekte sa počíta aj s rôznymi druhmi prechodov, ktoré sú popísané v kapitole 2. Záverom práce autorov je potvrdené, že použitie neurónových sietí má výrazný dopad na zlepšenie výsledkov detekcie prechodov pre chodcov. Signifikantné zlepšenie výsledkov však vzniklo na úkor času vyhodnocovania, ktorý bol vplyvom dodatočných kalkulácií predĺžený. Toto by sa ale podľa slov autorov dalo vyriešiť optimalizáciami, ktoré by tento vplyv znížili.



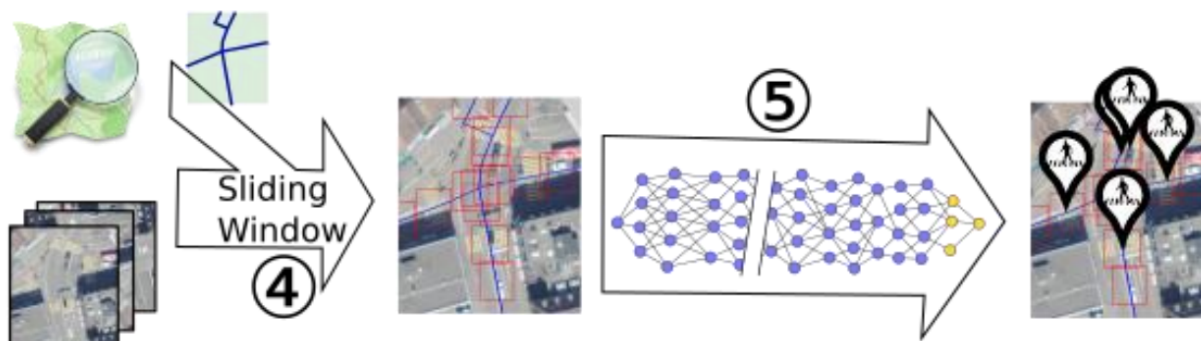
Obr. 3.4: Výstup použitím neurónovej siete s vyznačením reálnych hodnôt [4]

Na obrázku 3.4 je možné vidieť výstup algoritmu spracovaného v tejto publikácii. Červené a oranžové body reprezentujú pravú a ľavú stranu dát určených pre vykreslenie priamok. Modré priamky ukazujú výstup algoritmu a zelené priamky boli nakreslené ručne a označujú reálnu hranicu prechodu.

3.2.2 Detekcia prechodov v satelitných snímkoch použitím hlbokých neurónových sietí

Ďalším z možných prístupov, akým je možné sledovať pozíciu prechodov je využitie satelitných snímkov s nízkym rozlíšením. V kombinácii s už spomínaným systémom GPS by malo byť zabezpečené prispôsobenie rozlíšenia senzorov alebo aktivácia nových senzorov v oblasti, v ktorej sa prechod podľa dostupných dát má nachádzať. Práve detekciu prechodov z OpenStreetMap využíva projekt dostupný na ¹ s pridruženou prácou [26] z ktorej rozboru bude v tejto podkategórii vychádzané. V tejto publikácii autori predstavujú automatizačnú metódu na detekciu objektov zvolených tried, ktorá môže byť prispôbena pre rozličné druhy objektov. Podľa [26] metóda využíva tzv. hlboké konvolučné neurónové siete kde posledná vrstva predtrénovanej neurónovej siete je pretrénovaná špecifickými obrazmi a geografickými informáciami z OSM. Výsledkom je tak ľahko škálovateľná aplikácia pre detekciu objektov z leteckých snímkov.

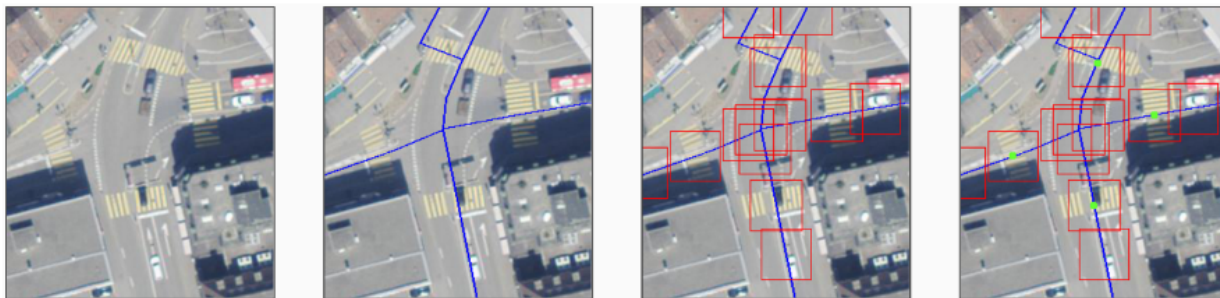
Ďalej je podľa [26] známy aj postup spracovania. Oblasť určená k analýze je systému predložená ako ohraničujúci objekt (angl. bounding box), ten je neskôr rozdelený na menšie objekty a je pridaný do fronty na spracovanie. Ak sa začne spracovávať požiadavka na vykonanie detekcie, z OSM sa získajú všetky ulice obsiahnuté v malom objekte. Ak sa v snímku nenachádzajú žiadne ulice, požiadavka na vykonanie je ukončená. Na obrázku 3.5 je možné vidieť graficky znázornený postup extrakcie dát a ich spracovanie. V bode č. 4 je použitá metóda tzv. sliding window pre získanie menších objektov, ktoré sú predložené na klasifikáciu DCNN.



Obr. 3.5: Postup detekcie z OSM [26]

¹<https://github.com/geometalab/OSMDeepOD>

V ďalšom obrázku 3.6 je možné vidieť ako sú vyhodnocované vstupy snímok z OSM. V prvej snímke je možné vidieť vstupný obrázok, ktorý je v ďalšej snímke doplnený o dáta o uliciach. Následne sú metódou sliding window z obrázku vystrihnuté obrázky o veľkosti 50x50 pixelov, čo je aj veľkosťou vstupných parametrov pre danú konvolučnú neurónovú sieť. Červené bounding boxy je možné vidieť na treťom a štvrtom obrázku. Týmto spôsobom sa dá znížiť počet overovaných obrázkov na oblasť ulice a vyhodnotiť tak pozíciu prechodu ktorá je na poslednom obrázku vyznačená zelenými bodmi.



Obr. 3.6: Spracovanie vstupného obrázku z OSM [26]

3.2.3 Lokalizácia prechodu založená na detektore Viola-Jones

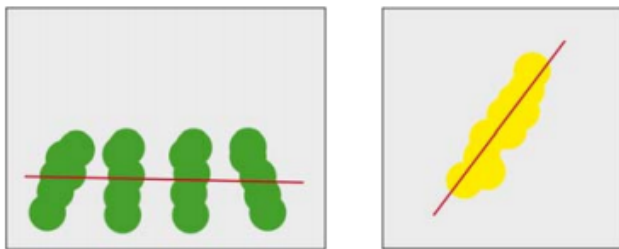
Použitie algoritmu Adaboost pre detekciu prechodov sa rozhodli využiť autori v článku [28]. Prístup je založený na princípe ktorý predstavili autori Viola a Jones [42] na použitie v kaskádovej architektúre. Adaboost je algoritmus ktorý tvorí zhľuky slabých klasifikátorov aby bol získaný finálny silný klasifikátor. Po zvolení jedného slabého klasifikátora, tréovacím vzorom sa zmenia váhy tak, aby boli označené aj príklady, ktoré ešte neboli klasifikované. Výhodou kaskádového trénovania je, že v iteratívnom vyhodnocovaní sú v skorých štádiách zamietnuté nevhodné výsledky, ktoré potom následne nie sú spracované v neskorších štádiách, čo má pozitívny vplyv na výpočtový čas algoritmu.

Pre detekciu prechodu a filtráciu javov, ktoré by mohli ovplyvniť správnosť výsledkov sú použité rozptyľové filtre ktoré normalizujú odchýlky kontrastu v obraze a dokážu tak odstrániť napr. presvetlené drážky vo vozovke ktoré môžu vykazovať vlastnosti prechodu.

Model natrénovaný na jednotlivé pruhy však podľa [28] na rozdiel od bežného použitia Adaboost na detekciu tváre reagoval nie len na celé čiary ale aj na ich časti. Kvôli tomu bolo navrhnuté zhľukovanie jednotlivých detekovaných oblastí zvolením presahov ohraničujúceho obdĺžniku. Ďalej boli zvolené 3 parametre pre klasifikáciu prechodu:

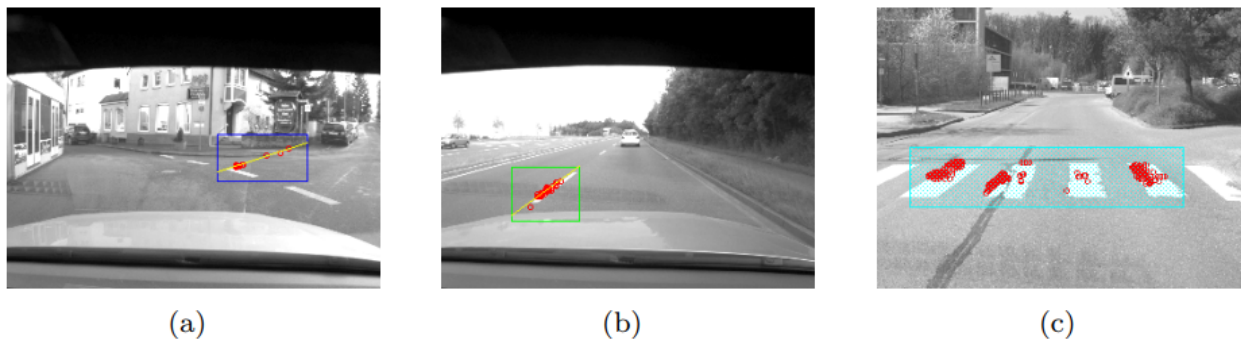
- počet výsledkov v rámci skupiny
- priemerná vzdialenosť medzi bodmi
- gradient regresnej priamky

Na obrázku 3.7 je možné vidieť zhlukovanie v prípade prechodu pre chodcov označené zelenými bodmi a regresnú priamku v prípade jednej čiary, ktorej body sú označené žltou farbou.



Obr. 3.7: Označenie regresnej priamky (červená) v prípade bodov nachádzajúcich sa v prechode (zelená) a bodov nachádzajúcich sa na jednej čiare (žltá) [28]

Naprieč obrazom boli zaznamenané rôzne prípady detekcie jednotlivých čiar. Počet bodov ktoré spadali do čiary sa častokrát javili ako čiara prechodu pre chodcov. Ak sa však v regresnej priamke ukázalo, že sa jedná o jednu čiaru, všetky body ktoré obsahovala boli zamietnuté. Niekedy sa však podľa autorov stalo, že vonkajšie časti prechodu vytvárali samostatné skupiny. Ak sa teda takáto skupina nachádzala vedľa detekovaného prechodu, jednotlivé skupiny boli spojené do jednej. Na obrázku 3.8 je možné vidieť detekciu jednotlivých čiar s nájdenými jednotlivými bodmi a ich zhlukovanie do skupín. V samostatnom obrázku (a) obrázku 3.8 je možné vidieť aj zlú detekciu odrazov spôsobenú pravdepodobne kvôli svetelným podmienkam a umiestneniu kamery pred čelným sklom.



Obr. 3.8: (a) normálna skupina, (b) skupina označené ako jedna čiara, (c) skupina označená ako prechod pre chodcov [28]

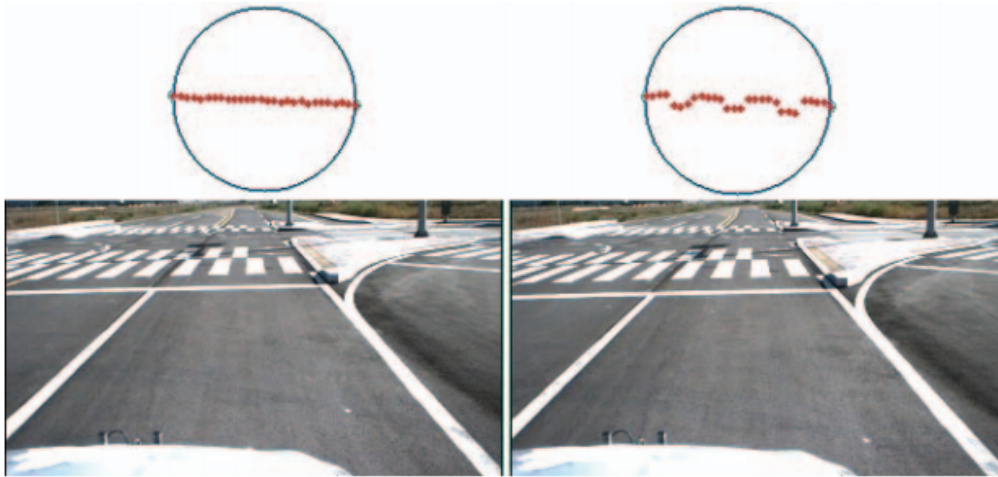
3.3 Metódy detekcie prechodov bez spracovania obrazu

Odhliadnuc od hlavnej témy tejto diplomovej práce a spracovaniu obrazu sa otvárajú nové možnosti detekcie prechodov, ktoré sú pre komplexne pochopenie problému a navrhnutia možných riešení taktiež veľmi dôležité. V kapitole nižšie je popísaný prístup bez spracovania obrazu s použitím laseru umiestneného na vozidle.

3.3.1 Detekcia laserom

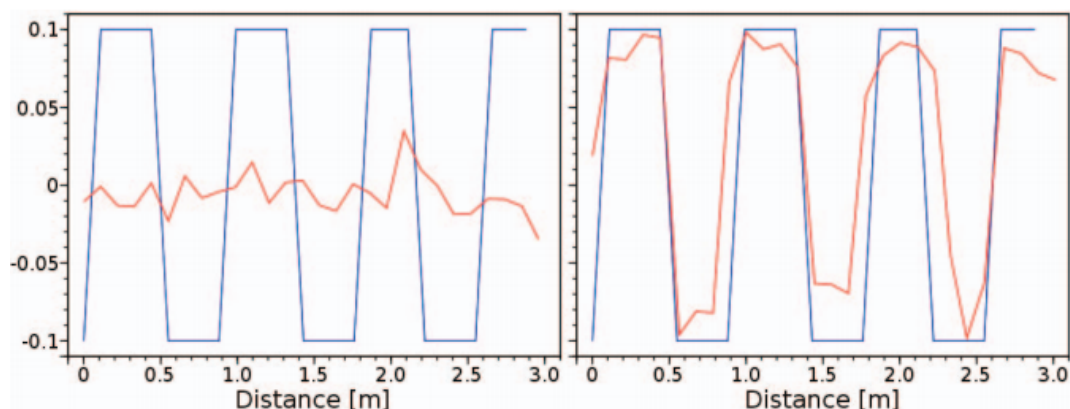
V článku v ktorom sa autori venujú detekcii prechodov pre chodcov je možné nájsť nový uhol pohľadu na danú problematiku. Autori vo svojom článku [20] spomínajú aj tradičnejšie prístupy popísané v tejto diplomovej práci ako napr. [18], [28] alebo v tejto diplomovej práci neobsiahnuté [14], [40]. Prichádzajú však s úplne novým prístupom ktorým je využitie reflektancie laseru na detekciu prechodu. Táto metóda využíva dobrú reflektanciu materiálu ktorý je bežne využívaný pre značenie prechodov.

Laser Measurement System - LMS (Systém merania laserom) bol namontovaný na vrchnú časť vozidla vo výške 1.7m a bol natočený o 7° nadol. Ak by laserový lúč narazil na uniformný povrch, vzdialenosť medzi povrchom vozovky a vozidlom je približne 12.9m vzhľadom na povrch vozovky a 13.21m vzhľadom na smer laseru. Pri jazde vozidlom je z laseru upevneného na vozidle získavané dáta a je vytváraný vektor $CMD = \{x_j, y_j | j = 1, \dots, n\}$. Vektory ukladané do CMD sú naznačené v obrázku 3.9. V ideálnom prípade by dáta o prechode mali ležať na povrchu vozovky. V reálnom svete je však jazda ovplyvnená rôznymi faktormi ktoré by toto snímanie mohli narušiť. Navyše v rôznych krajinách existujú rôzne druhy prechodov a tak je potrebné algoritmus prispôsobiť pre rôzne šírky, medzery medzi čiarami a dĺžky prechodu.



Obr. 3.9: Výstup z laseru [20]

Práve kvôli rozličnej šírke a vzdialenosti medzi čiarami, perióda P ktorá predstavuje cestu od ľavého dolného rohu po ďalší ľavý dolný roh v priebehu na obrázku 3.10 je zvolená v rozmedzí hodnôt 0.9m - 1.2m. Na tomto obrázku je taktiež možné vidieť aj označenie ideálneho priebehu zaznačeného modrou farbou a priebehu vstupných dát z laseru v ľavom grafe bez prítomnosti prechodu a v pravom grafe s prechodom prítomným na vozovke. Amplitúda vstupu striktne závisí na vstupných dátach laseru.

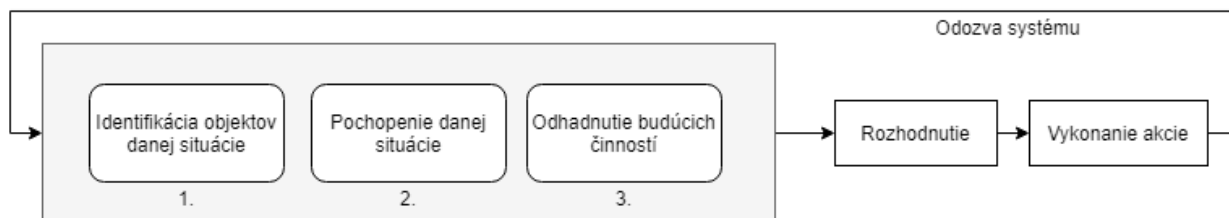


Obr. 3.10: Obdĺžnikový priebeh (modrá) a vstupný signál (červená) [20]

Záverom autori popisujú ideálne použitie pre spracovanie v reálnom čase s rýchlosťou spracovania v priemere 0.18ms čo vykazuje vhodnosť použitia tohoto prístupu nie len pre detekciu prechodov ale aj pre detekciu čiar na ceste. Bližšie informácie k detekcii čiar pomocou laseru v projekte [21].

3.4 Využitie detekcie chodcov vzhľadom k lokalizácii prechodu

Neoddeliteľnou súčasťou snímania prechodov pre chodcov je aj snímanie a lokalizácia chodcov, ktorí by prechod eventuálne chceli použiť. Z hľadiska autonómnych vozidiel je táto téma blízko spojená s detekciou prechodov a má zmysel zamyslieť sa nad problematikou detekcie prechodov zo širšieho spektra. Rovnaký názor majú aj autori práce [36], ktorí sa v nej venujú práve správaniu sa chodcov pri prekonávaní cestných komunikácií. Na obrázku 3.11 je možné vidieť 3 levely k uvedomeniu si situácie, ktoré musia byť v oblasti autonómnych vozidiel zavedené.



Obr. 3.11: Diagram nutnosti uvedomenia si situácie podľa [36]

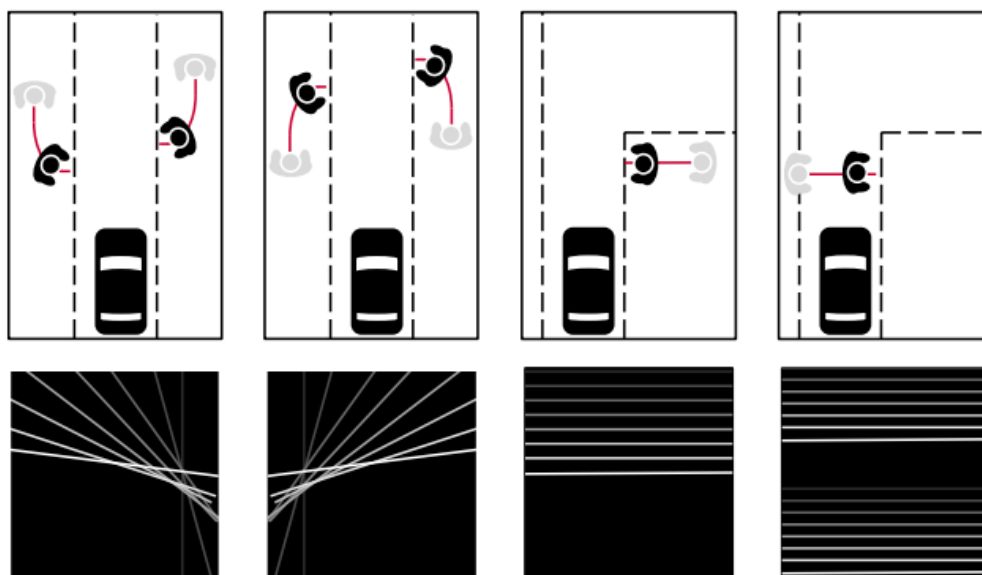
V prípade chodcov je nutné spracovať predikciu v limite 0,5 až 1,5 sekundy. Aktívne bezpečnostné systémy, ktoré by mali tieto podmienky aplikovať však môžu reagovať iba na chodcov, ktorí sú na ulici alebo sa na ňu chystajú vstúpiť. Riešením problému uvádzajú autori zlepšenia v bode 2. obrázku 3.11. Úmysel je podľa [12] definovaný ako príčina ľudskej akcie a jeho detekcia umožňuje projekciu budúcej situácie pomocou priemerovania predpokladaných akcií. V oblasti autonómnych vozidiel ide hlavne o úmysel prekonať cestnú komunikáciu. Klasifikáciou chodcov do skupín podľa

úmyslu prekonať cestnú komunikáciu chodcov rozdeľuje do dvoch skupín. Vďaka predikcii chovania chodcov tak autonómny systém vozidla dokáže včas vyhodnotiť bezpečnostné riziká, prispôbiť jazdu či dokonca komunikovať s chodcom. Štúdia [36] je zameraná na detekciu zámerov chodcov ktorí majú nepriamy styk napr. s cestným značením akým sú prechody, autobusové zástavky a okraje ciest. Pomocou vektoru príznakov je definovaný algoritmus ktorý popisuje pohyb chodca relatívne k spomínaným označeniam a oblastiam vozovky, prípadne inej cestnej komunikácie.

Kohler v [25] použil MCHOG deskriptor v kombinácii s SVM klasifikáciou pomocou čoho určuje, či chodci budú pokračovať v čakaní na okraji alebo začínajú s pohybom vpred. Keller v [23] pomocou pozičných parametrov a optického toku zistil, či chodec bude pokračovať v chôdzi alebo zastaví pred okrajom vozovky.

V referenčnom článku [36] bol navrhnutý systém, v ktorom bola integrovaná detekcia chodcov a čiar čo bolo vstupom pre detekciu navrhovaným algoritmom. Prístup je zložený z dvoch hlavných krokov. V prvej časti je získaný vektor príznakov pozostávajúci z dvoch elementov: (a) popis vzoru pohybu chodca v závislosti na relevantom značení ciest, ktoré je vyhodnocované pomocou segmentácie obrazu a (b) popis priestorového rozloženia elementov ako sú napr. prechody, zástavky a nástupové ostrovčeky. Zároveň je výsledný vektor spojený s vektorom, ktorý obsahuje informácie reprezentujúce napr. obsadenie zástavky alebo počet ľudí čakajúcich pred prechodom pre chodcov. V druhej fázi je vektor príznakov klasifikovaný pomocou SVM.

Pohyb chodcov je vyhodnocovaný pomocou CMHI bližšie popísaného v referenčnom článku [36]. Na obrázku 3.12 je možné vidieť reprezentáciu CMHI pre vyznačený pohyb chodcov. Šedou je zaznačený začiatok pohybu a červená krivka označuje smer pohybu.



Obr. 3.12: CMHI pre vyznačený pohyb chodcov podľa [36].

Záverom tohto projektu bolo zistené, že systém si je takmer istý u približne 90% prípadov a

veľkú dávku neistoty vykazuje najmä v prípadoch, kedy chodec kráča rovnobežne s cestou a začína otáčať hlavu smerom k blížiacej sa premávke. Pohyb hlavy je teda dôležitým indikátorom, ktorý môže bližšie určiť zámery chodca.

3.5 Porovnanie použitia rôznych prístupov

Ako vyplynulo z prehľadu existujúcich riešení, pre problematiku detekcie prechodov pre chodcov je možné použiť rôzne techniky spracovania obrazu. Hlavným rozdielom medzi metódami s tréновaním a bez tréновania je nutnosť prípravy a vytvorenia datasetu. V tejto diplomovej práci bude navrhnutý detektor použitím prístupu bez tréновania pomocou knižnice OpenCV ktorý po následnom spracovaní dát otvorí nové možnosti s použitím neurónových sietí. Ďalej budú popísané hlavné črty jednotlivých prístupov a problémy, ktoré pri ich použití vznikajú.

3.5.1 Použitie prístupu bez tréновania

Pre extrakciu požadovaných vlastností z obrazov je možné použiť rôzne dostupné knižnice pre spracovanie obrazu (napr. OpenCV, SimpleCV, scikit-image a iné). S použitím prístupu bez tréновania sú hlavné problémy najmä s kvalitou obrazu. Pri nevhodnom umiestnení kamery pri snímaní vozovky môže dôjsť k rôznym nežiadúcim efektom ako je napríklad odrážanie slnka od čelného skla a presvetlený obraz. Ďalším z problémov je snímanie v rôznych obdobiach dňa, prípadne počas nočnej jazdy s čím súvisia aj aktuálne poveternostné podmienky a stav vozovky napr. počas silného dažďa alebo zlej viditeľnosti spôsobenej hmlou. V prípade filtrovania prechodu pre chodcov v obraze hrá veľkú rolu aj amortizácia prechodu, ktorá môže zapríčiniť splývavosť s vozovkou. Prechody pre chodcov sú v oblastiach bydliska autora tejto diplomovej práce vo veľmi zlom stave (viď 3.13) a častokrát sú počas testovania v tejto diplomovej práci nesprávne vyhodnotené navrhnutými algoritmami.



Obr. 3.13: Ťažko identifikovateľné prechody.

3.5.2 Použitie neurónových sietí

Neurónová sieť je zložená z jednoduchých neurónov spôsobom, kedy výstup jedného z neurónov môže byť vstupom iného neurónu [31]. Učenie pri neurónových sieťach je možné rozdeliť na tzv. strážené (z angl. supervised) a nestrážené (z angl. unsupervised). [4]. Unsupervised algoritmy nemajú fixne označené výstupy ale snažia sa ich zhlukovať podľa spoločných vlastností. Príkladom môže byť napr.

oddelenie ľudskej tváre od pozadia zhlukovaním vlastností tváre. Supervised algoritmy sú naopak vhodné, ak máme výstupy siete presne definované.

V spojení s neurónovými sieťami sú v dnešnej dobe najväčším problémom tréningové dáta a veľký výpočtový výkon, ktorý je potrebný na natréningovanie modelu ktorý je následne využitý pre vyhodnotenie vstupu. Ďalším problémom je aj pomerne veľký výpočtový výkon a čas, v ktorom sa evaluácia vykoná. Pre spracovanie výsledkov sa častokrát využíva spracovanie veľkým výpočtovým výkonom na serveri a následný prenos informácií k odosielateľovi požiadavku. V tejto diplomovej práci bude predpokladané využitie tohoto scenáru, nakoľko vyhodnotenie výsledku má od ideálneho spracovania v takmer reálnom čase (rádovo desiatky až stovky milisekúnd) veľmi ďaleko a výsledok je spracovaný až v jednotkách sekúnd. Pre potreby tejto diplomovej práce bude použitý konkrétny typ neurónovej siete Mask-RCNN.

Z prieskumu dostupných obdobných riešení je zrejmé, že v rôznych prípadoch je vhodné použiť spracovanie obrazu ako pomocou prístupov bez tréningovania (napr. použitím knižnice OpenCV), tak aj využitím neurónových sietí a tréningom modelov pomocou dostupných alebo vlastných datasetov. Prieskum dostupných riešení zároveň ukázal, že táto téma v dnešnej dobe získava väčšiu popularitu a s príchodom nových technológií a nových prístupov je možné riešiť ju niekoľkými rôznymi spôsobmi. Niektoré z prístupov získaných v prehľade dostupných riešení budú použité aj v tejto diplomovej práci.

Kapitola 4

Vlastné riešenie

Na základe prehľadu existujúcich riešení dostupných v kapitole 3 bol vyvođený postup návrhu a implementácie detektoru prechodov pre chodcov. Návrh pozostáva z použitia dvoch samostatných prístupov popísaných v porovnaní prístupov vyplývajúcich z prehľadu existujúcich riešení v kapitole 3.5. V popise vlastného riešenia je zdokumentovaný návrh a tvorba detektoru pomocou prístupu bez tréovania s použitím knižnice OpenCV ktorý sa nachádza v sekcii 4.1.1 a natréovanie modelu neurónovej siete použitím knižnice Tensorflow a M-RCNN ktoré je možné nájsť v sekcii 4.2.3. Vlastné riešenie teda pozostáva z vytvorenia dvoch samostatných algoritmov ktoré slúžia k detekcii prechodov pre chodcov. Algoritmus bez tréovania bude primárne využitý pri tvorbe datasetu a sekundárne bude zaradený aj pri testovaní prístupu s tréovaním ako komplementárny prvok pre správne určenie prechodu pre chodcov. S použitím prvého prístupu bez tréovania úzko súvisí vytvorenie datasetu, ktorý je dôležitý pre správne natréovanie neurónovej siete. Nakoľko dataset explicitne určený na tréovanie prechodov pre chodcov nebolo možné nájsť, musel byť použitý algoritmus, ktorý z dostupných datasetov cestných komunikácií dokáže aspoň približne určiť, či sa v danom obrázku prechod nachádza. V kapitole 4.1.2 je popísaná tvorba datasetu a čitateľ bude oboznámený s použitými technológiami pre tvorbu datasetu, vytváranie skriptov a natréovanie modelu pomocou neurónovej siete. Ďalšie podkapitoly sa teda budú venovať návrhu algoritmov, tvorbe datasetu, predspracovaniu obrazu a tréovaniu modelu neurónovej siete.

4.1 Návrh algoritmu bez tréovania a tvorba datasetu

Pre splnenie úlohy je potrebné rozdeliť problém na niekoľko samostatných častí. Najskôr je potrebné získať dáta pre tréovanie neurónovej siete. Po extrakcii dát zo vstupných videí a datasetu nasleduje ručná korekcia, filtrovanie a následná anotácia dát. Po vykonaní všetkých potrebných krokov môže byť zahájená posledná fáza vytvorenia detektoru ktorej súčasťou je samotné natréovanie a otestovanie na zvolených datasetoch.

4.1.1 Vytvorenie detektoru bez tréovania

Postup pre vytvorenie detektoru začína vytýčením oblasti záujmu pre pozíciu vozovky vzhľadom k záberu kamery. Následne je oblasť zmenšená na jednotnú veľkosť kvôli urýchleniu procesu detekcie algoritmu určenému pre filtráciu dát do vytváraného datasetu. Ďalším krokom je filtrácia obrazu kvôli rôznym nedokonalostiam a amortizácií prechodu. V prípade nepoužitia filtrácie sa po hľadaní kontúr prechodu stávajú výsledky pomerne nepresné. Po vykonaní prahovania vo farebnom formáte HSV je vyfiltrovaná biela farba (resp. odtiene šedej) ktoré sú najčastejšie zastúpené v prechodoch na území Českej a Slovenskej republiky.

V rámci tejto práce má pomerne veľké zastúpenie technológia OpenCV. OpenCV je open source knižnica napísaná v jazyku C a C++. Aktívny je rovnako aj vývoj na rozhraniach pre jazyky Python, Ruby, Matlab a iné [7]. Táto technológia v tejto diplomovej práci nebude bližšie popísaná, nakoľko existuje množstvo publikácií v ktorých má čitateľ možnosť dozvedieť sa (napr. [7], [6], [27]). Ďalšie použitie tejto knižnice bude vzťahované výhradne na účely implementácie riešenia. Pre prácu v projekte je využívaný balíček opencv-python pre jazyk Python.

Hranice HSV sú určené podľa pomocného skriptu, v ktorom bolo pomocou posuvníkov možné nastaviť úroveň jednotlivých hodnôt tak, aby boli dosiahnuté čo najlepšie výsledky z filtrácie prechodu pre chodcov. Ukážku nastavovania je možné vidieť na obrázku 4.1.



Obr. 4.1: Ukážka hľadania HSV prahovania - obrázok pred a po aplikácii prahovania

Po nájdení kontúr nasleduje použitie Houghovej transformácie (viď. [3]). V tomto riešení je použitá verzia probabilistickej Houghovej transformácie ktorá by mala poskytovať úspornejšiu implementáciu Houghovej transformácie a v balíčku OpenCV je nájdeme pod názvom *HoughLinesP*. Po nájdení obrysov sa ďalej skúmajú parametre, ktoré boli ručne nastavené sledovaním opakujúcich sa vzorov v obrazoch, v ktorých sa prechod pre chodcov nachádzal. Výstupom tohto algoritmu je množina dát, na ktorých sa nachádza prechod, množina vytýčených oblastí, v ktorých sa nachádza prechod a množina vytýčených oblastí, v ktorých sa prechod nenachádza. Konkrétne nastavenie parametrov dôležitých krokov je popísané nasledujúcim postupe. Ako prvé je vykonaná mediánová filtrácia kvôli potlačeniu nedokonalostí prechodu. Následne je použité prahovanie vo formáte HSV, ktorého výsledok je vyfiltrovaný pomocou Cannyho detektoru hrán. Po nájdení kontúr pokračuje

parametrická filtrácia na základe určených parametrov prechodu. Porovnávané parametre sú obvod pre jednotlivé pruhy prechodu spolu s uhlom, ktorý zvierajú priamky výstupu použitia Houghovej transformácie, ktorá je ďalším z krokov popisovaného algoritmu.

Nakolko bolo na testovacích dátach vykonané adaptívne prahovanie a jeho rôzne varianty no žiadna z nich neposkytla dostatočne vhodné výsledky, pre potreby tejto diplomovj práce bol navrhnutý jednoduchý postup, ktorý by dokázal z rôznych podmienok nasnímaného obrázku a rôznych odtieňov bielej fraby, ktorá je primárne použitá pre značenie prechodov určiť spodnú hranicu zložku VALUE vo formáte HSV. Tento postup bol vybraný na základe experimentálneho testovania z dôvodu lepších výsledkov testovacích dát v porovnaní s adaptívnym prahovaním. Pomocou náhodne vybraných obrázkov boli extrahované dáta potrebné pre vytvorenie regresnej priamky. V realite častokrát biela farba prechodu degraduje na rôzne odtiene šedej a pri zlej viditeľnosti taktiež nemusí byť prechod v obraze v porovnaní z pozadím dostatočne kontrastný. Pre prispôbenie citlivosti snímání prechodu boli preto v náhodných obrázkoch odčítané hodnoty mediánu zložky VALUE formátu HSV a podľa pomocného skriptu boli nastavené hraničné hodnoty pre danú hodnotu mediánu. Výsledkom je rovnica

$$V_{new} = 2.02 * x - 90.17 \quad (4.1)$$

kde x je medián zložky VALUE prehľadávaného obrázku a V_{new} je identifikovaná spodná hranica prahovania vo formáte HSV.

Testovaním sa však zistilo, že pri vysokých mediánoch hodnôt VALUE častokrát vychádzajú po dosadení do rovnice nepostačujúce výsledky, čo je spôsobené odfiltrovaním veľkej časti farebného spektra a následným odstránením prechodu z obrazu. Testovaním sa zistila ideálna minimálna hodnota, ktorú nesmie spodná hranice zložky VALUE prekročiť. Celý kód definovania prahovania HSV je možné vidieť v nasledujúcom odstavci:

```
hsvImage = cv2.cvtColor(inputImage, cv2.COLOR_BGR2HSV)
v_upper = min(210, int(2.01 * cv2.meanStdDev(hsvImage)[0][2][0] - 90.17))

lower = np.array([20, 0, v_upper])
upper = np.array([180, 100, 247])
mask = cv2.inRange(hsv, lower, upper)
```

Výpis 4.1: Ukážka nastavenia a aplikácie prahovania v načítanom obrázku

Konečným výsledkom vyhodnotenia podľa navrhnutého algoritmu je príznak, či sa prechod vo vstupnom obraze nachádza alebo nie. Tento príznak bol ďalej kvôli testovaniu v kapitole 5.3 doplnený aj o určenie pozície prechodu pre chodcov v obraze. Príznak, či sa prechod v obraze nachádza je však pre aktuálne potreby vytvorenia datasetu pre natrénovanie neurónovej siete dostatočujúci. Tvorba datasetu je ďalej popísaná v kapitole 4.1.2.

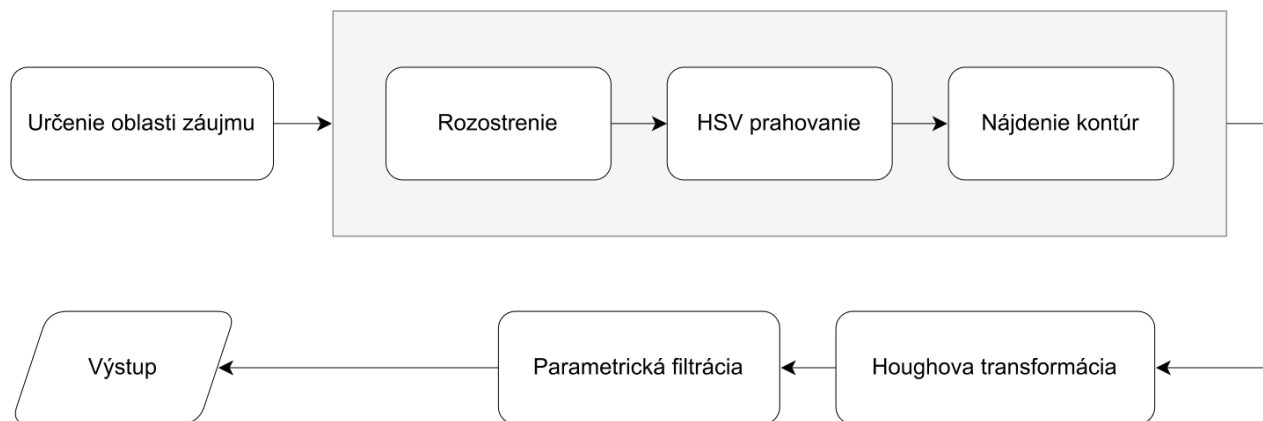
4.1.2 Tvorba datasetu

Pre čo najpresnejšie natrénovanie neurónovej siete je potreba mať kvalitný a rozsiahly dataset. Dataset môže byť definovaný ako zhluk obrázkov popisujúcich jeden jav alebo objekt. Dáta môžu byť anotované, pokiaľ sa v danom datasete požadované objekty nachádzajú na konkrétnej pozícii.

Pre potreby tejto diplomovej práce bol použitý dataset IARA nachádzajúci sa na¹ spolu s vlastným datasetom. Dataset IARA obsahuje záznamy jazdy vozidla a pre účely tejto diplomovej práce je nutné jeho ďalšie spracovanie. Dataset obsahuje 12,441 obrázkov rozdelených do štyroch samostatných sekvencií nahrávaných v Espírito Santo, Vitória. Pozícia kamery umiestnenej na aute sa nemení, preto je dataset vhodný pre vytýčenie regiónu záujmu.

Ďalší z použitých zdrojov bol poskytnutý vedúcim tejto diplomovej práce a pozostáva z 94 videí zachytených v oblasti mesta Ostrava. Kamera snímajúca cestu bola umiestnená za čelným sklom vozidla, preto sú niektoré snímky skreslené najmä odrazmi slnka. Videá sú zaznamenané počas rôznych častí dňa, za rôznych podmienok a premenlivého počasia, čo je pre tréovanie neurónovej siete ideálne. Z videí, ktorých celková dĺžka presiahla 2 hodiny boli extrahované snímky ktoré boli následne spracované pomocou navrhnuté algoritmu v sekcii 4.1.1. Video bolo natáčané frekvenciou 30 FPS a pre zachovanie čo najväčšieho množstva dát a zároveň odfiltrovanie podobných duplikátov bola zvolená extrakcia každej štvrtej snímky, aby sa snížil počet kontrolovaných snímok ale aby zároveň boli zachytené aj objekty pri vyššej rýchlosti vozidla.

V oboch prípadoch datasetov bolo potrebné vykonať filtráciu a anotáciu dát pre potreby vytvorenia neurónovej siete.



Obr. 4.2: Diagram postupu detekcie pomocou OpenCV

Pre filtráciu potencionálne vhodných dát na tréovanie bol využitý detektor prístupu bez tréovania popísaný v kapitole 4.1.1, ktorý bol navrhnutý pomocou knižnice OpenCV v jazyku Python

¹<https://github.com/rodrigoberriel/streetview-crosswalk-classification/tree/master/datasets>



Obr. 4.3: Ukážka výstupu algoritmu pre vytvorenie datasetu

Na obrázku 4.3 je možné vidieť časť výstupných obrázkov, ktoré boli extrahované pomocou vytvoreného algoritmu. Vo výstupe sa nachádzajú aj falošné pozitívne výsledky, ktoré boli často-krát znásobené napr. dlhším pohybom za vozidlom, ktoré skreslenie zapríčiňovalo alebo občasné odrazy slnka od čelného skla vozidla. Detailnejší popis extrakcie dát spolu s časom potrebným na vyhodnotenie a počtami výsledkov je možné nájsť v tabuľke 4.1.

Tabuľka 4.1: Porovnanie extrakcie dát z datasetov

Názov	Počet obrázkov	Čas extrakcie	Výsledných obrázkov celkovo (P/N)
Vlastný	12,441	11,514s	601 (342/259)
IARA	115,525	569s	183 (145/38)
Celkovo	127,966	12,083s	784 (497/297)

Využitie tohoto algoritmu sa kvôli veľkému množstvu falošných pozitívnych výsledkov javí pre reálne použitie ako nevhodné, pre extrakciu a vyfiltrovanie časti dát však úplne postačujúce. Pre využitie tohoto algoritmu bude overená jeho vhodnosť pri testovaní, kde môže byť ďalším pomocným faktorom pri určení pravdivosti výsledku.

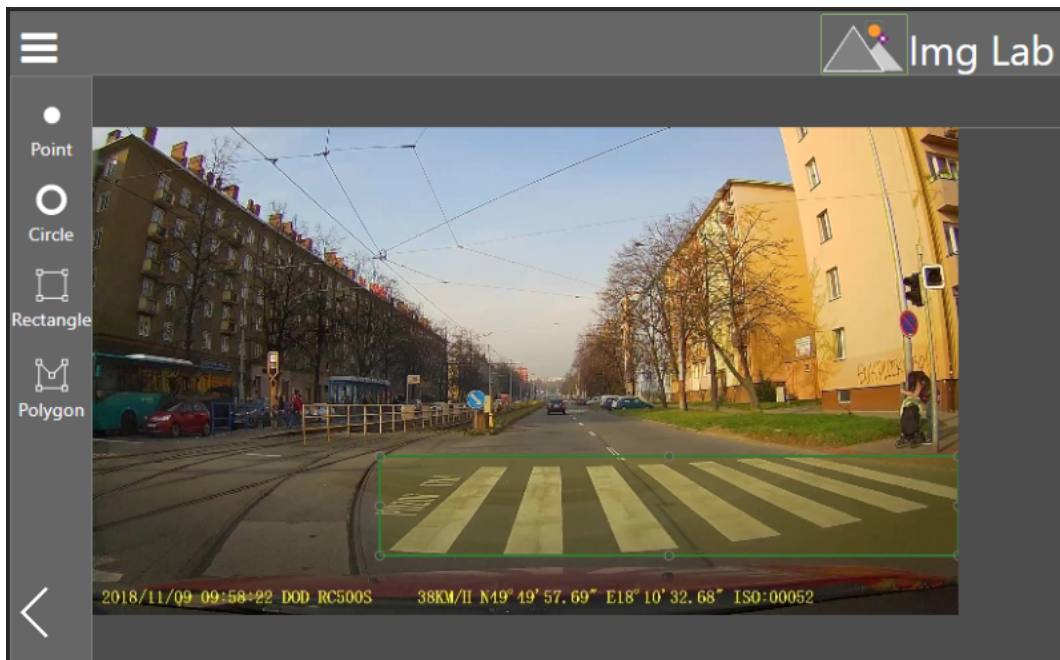
4.1.3 Anotácia dát

Pre spracovanie detektoru prechodov boli použité technológie stručne popísané v nasledujúcich podkapitolách. Okrem nižšie spomenutých technológií boli ďalej použité pomocné skripty písané najmä v jazyku Python a iné pomocné nástroje pre zjednodušenie práce. Pre vývoj samotnej aplikácie bolo použité vývojové prostredie od spoločnosti JetBrains PyCharm, IDE pre jazyk Python ktoré je pre edukatívne účely voľne dostupné na stiahnutie na oficiálnych stránkach ².

²<https://www.jetbrains.com/pycharm/>

Imglab

Imglab je anotačný webový klient ktorého zdrojový kód je možné nájsť na adrese ³ a samotná služba je dostupná na webovej adrese⁴. Tento anotačný nástroj slúži pre predspracovanie dát potrebných pre tréovanie neurónovej siete. Nástroj, ktorého používateľské prostredie je možné vidieť na obrázku 4.4, po importe súborov určených k anotácii umožňuje vďaka prívetivému používateľskému rozhraniu jednoduchú a rýchlu anotáciu dát. V prípade tejto diplomovoej práce išlo o označenie pozície, kde sa nachádza prechod pre chodcov a dodatočne aj označenie vertikálnych značiek prechodov pre chodcov kvôli návrhom na zlepšenie existujúceho algoritmu.



Obr. 4.4: Príklad anotácie pomocou služby imglab.in

Samotný nástroj umožňuje export napr. pre tieto technológie:

- dlib XML
- dlib pts
- Pascal VOC
- COCO

V tejto práci využívaná a taktiež ďalej popisovaná technológia Tensorflow zatiaľ svoje miesto v tomto zozname nenašla, aj keď autori uvádzajú že plánujú jej doplnenie. Avšak aj v prípade menšej

³<https://github.com/NaturalIntelligence/imglab>

⁴<https://imglab.in>

úpravy a prispôsobenia vygenerovaného XML súboru nástroj Imglab poskytuje značné zjednodušenie práce. Kvôli použitiu vyfiltrovaných obrázkov pre tréovanie siete R-CNN popísanej v sekcii 4.2 je potrebné, aby bolo vo vstupných dátach vyznačené, kde sa nachádza pozícia prechodov pre chodcov. Pre túto úlohu bol použitý nástroj Imglab 4.1.3 ktorý je dostupný ako webový nástroj a slúži práve pre anotáciu dát. Výstupný formát síce nepodporuje používanú technológiu Tensorflow avšak výstup XML pre Dlib je pomerne jednoducho prevediteľný pre prácu s Tensorflow.



Obr. 4.5: Ukážka anotovaných dát v službe Imglab

Na obrázku 4.5 je možné vidieť priebeh a označenie anotovaných dát. Výstupný formát súboru XML obsahuje jednotlivé názvy obrázkov s definovanými ohraničujúcimi boxami, ktorých ľavá dolná pozícia a šírka spolu s výškou sú definované v XML súbore.

Vo vytvorenom datasete bolo anotovaných 531 prechodov v 497 vstupných obrázkoch ktoré boli použité pre tréovanie neurónovej siete.

4.2 Návrh algoritmu pre natréovanie modelu pomocou neurónových sietí

Výstup algoritmu pre vytvorenie datasetu s definovanou oblasťou záujmu otvoril možnosť pre implementáciu neurónovej siete, ktorá objekt vo vstupnom obrázku indentifikuje a určí jeho polohu. V tomto prípade je teda overená účinnosť tréovania v prípade, kedy bude na vstup tréovania použitá vydefinovaná oblasť záujmu pripravená v kapitole 4.1.2. V tejto kapitole budú ďalej bližšie popísané použité nástroje pre vytvorenie a natréovanie neurónovej siete v sekcii 4.2.1. V ďalšej časti 4.2.2 bude popísaná použitá neurónová sieť a v sekcii 4.2.3 bude natréovaná pomocou datasetu vytvoreného v časti 4.1.2.

4.2.1 Použité nástroje

Pre vytvorenie skriptu a natrénovanie modelu pomocou neurónovej siete bola použitá cloudová služba Google Colaboratory popísaná v kapitole 4.2.1 a služba Google Street View pre vytvorenie ďalšieho testovacieho datasetu

Tensorflow a Keras

TensorFlow je systém strojového učenia, ktorý je využívaný vo veľkom merítku hlavne v heterogénnych prostrediach. Knižnica Tensorflow je vyvíjaná spoločnosťou Google a je nástupcom systému DistBelief, ktorý Google používal na tréňovanie neurónových sietí od roku 2011 [1]. V systéme Tensorflow sa všetky vstupné dáta nazývajú Tensory. Tie je možné predstaviť si ako n-dimenzionálne polia obsahujúce primitívny dátový typ ako je napr. int32 alebo float32. Pre tréňovanie neurónovej siete a následnú evaluáciu je v tejto diplomovej práci použitý práve systém Tensorflow. Ten môže byť spúšťaný na rôznych výpočtových jednotkách ako je napr. CPU, GPU alebo TPU. Využitie pre túto diplomovú prácu bude ďalej popísané v ďalších kapitolách.

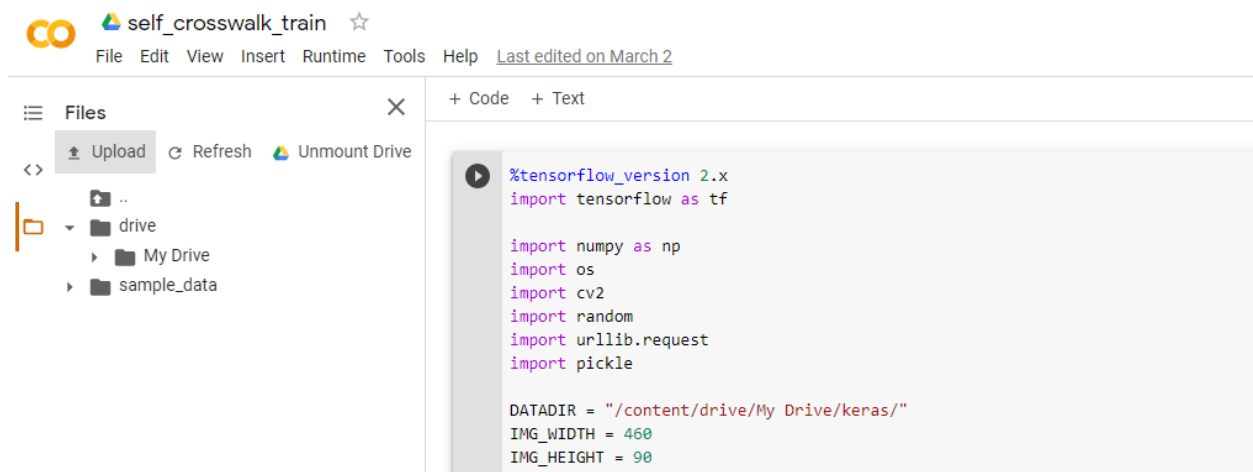
Pre tvorbu neurónových sietí je použitý Keras, modulárna knižnica ktorá je postavená na technológií Tensorflow [17]. Keras umožňuje rýchle a jednoduché vytvorenie neurónovej siete a ponúka množstvo komponentov pre prácu s datasetom. Konkrétne použitie tejto knižnice bude ďalej uvedené.

Google Colaboratory

Google Colaboratory, taktiež známe ako Colab (označenie Colab bude ďalej využívané v texte tejto diplomovej práce), je cloudová služba založená na Jupyter Notebooks. Poskytuje prostredie pre hlboké učenie a bezplatný prístup k výkonným GPU [8]. Tento fakt ukazuje aj možnosť využitia pre iné GPU-centricke operácie. V cloudovej službe Colab je podľa [8] použitý nasledujúci hardvér:

- Procesor Intel Xeon (bližšie nešpecifikovaný)
- 13 GB RAM
- Pascal VOC
- GPU Tesla K80 (12GB) s 2496 CUDA jadrami

V práci [8] bol vykonaný test klasifikácie obrazu pomocou Google Colab a bola úspešne natrénovaná konvolučná neurónová sieť pre dataset ručne písaných čísl MNIST. Tento experiment potvrdzuje vhodnosť využitia tejto platformy pre vytváranie riešenia tejto diplomovej práce. Na obrázku 4.6 je možné vidieť používateľské prostredie služby Colab s pripojenou službou Google Drive a inicializovaným listom na báze Jupyter Notebooks.



Obr. 4.6: Používateľské prostredie Google Colab

Google Street View

Pre vytvorenie spoľahlivého detektoru je jedným z najdôležitejších faktorov kvalita a veľkosť datasetu. V roku 2007 bola spustená Google Street View ako súčasť Google Maps a od tej doby rapídne expandovala [15]. V tejto práci [15] je taktiež ozrejmene aj použitie techník na rozmazávanie tvárí a evidenčných čísel vozidiel. Na ochranu súkromia osôb sa v posledných rokoch začal klásť omnoho väčší dôraz. Tento problém je premietnutý aj do oblasti tvorby neurónových sietí, keďže s touto tvorbou súvisia rôzne datasety ktoré môžu odhaľovať citlivé informácie o osobách. V tejto diplomovej práci sú pre účel testovania použité filtrované dáta extrahované zo služby Google Street View, vďaka čomu je pre potreby prezentácie výsledkov vyriešený problém s ochranou osobných údajov.

4.2.2 Neurónová sieť Mask-RCNN

Implementácia neurónovej siete Mask-RCNN [2] je navrhnutá v jazyku Python za pomoci knižníc uvedených v 4.2.1 Keras a Tensorflow a je dostupná pod prislúchajúcou citáciou [2] v ktorej sú uvedené vlastnosti a použitie neurónovej siete. Metóda nazývaná Mask R-CNN rozširuje Faster R-CNN a je jednoducho rozšíriteľná o ďalšie vlastnosti detekcie [19]. Je pomocou nej možné detekovať viaceré druhy objektov. Model generuje ohraničujúce obdĺžniky pre každú inštanciu hľadaného objektu v obraze. Je založený na FPN a ResNet101 o ktorých bližšie informácie sú dostupné v prislúchajúcich citáciach [29] a [19].

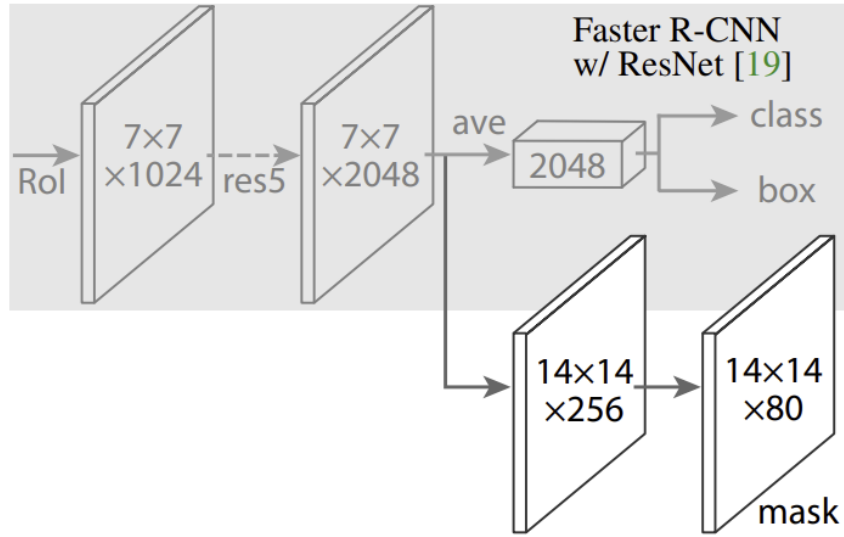
Podľa [19] autori navrhli Mask R-CNN z pôvodnej Fast R-CNN tak, aby používala rovnaké procedúry pre vyhľadanie triedy a ohraničujúceho obdĺžniku hľadaného objektu a zároveň vyhľadáva binárnu masku pre RoI. Autori ďalej aplikujú paralelné spracovanie klasifikácie ohraničujúceho obdĺžniku a regresiu, čo má za následok signifikantné zjednodušenie viac-krokového postupu.

Počas tréningu autori formálne definujú multi-triedovú stratu každého RoI definovanú ako

$$L = L_{cls} + L_{box} + L_{mask} \quad (4.2)$$

kde L je celková strata, L_{cls} a L_{box} sú rovnaké ako v [16]. Maska má Km^2 dimenzionálny výstup každého RoI, ktorý kóduje K binárnych másk rozlíšenia $m \times n$, na čo je aplikovaný per-pixel sigmoid a definuje L_{mask} ako priemernú binárnu cross-entropy stratu. Definícia L_{mask} autorov umožňuje sieti generovať masky pre každú triedu bez vzájomného ovplyvňovania tried. Autori zároveň definujú vypustenie kvantifikácie hraníc RoI a používajú bilineárnu interpoláciu pre výpočet príznakov na štyroch oblastiach každého RoI.

Ako hlavnú vnútornú sieť autori definujú použitie ResNet o hĺbke 50 alebo 101 vrstiev. Pre konkrétnu implementáciu podľa [2] je použitá ResNet101. Na obrázku 4.7 je možné vidieť rozšírenie ResNet o masku. Čísla označujú priestorové rozlíšenie a kanály, šípky označujú konvolučné, dekonvolučné alebo FC vrstvy. Všetky konvolučné filtre sú o veľkosti 3×3 s výnimkou výstupného konvolučného filtra o veľkosti 1×1 . Dekonvolučné filtre majú veľkosť 2×2 s posunom o veľkosti 2. V skrytých vrstvách je použitá aktivačná funkcia ReLu. Schéma na obrázku 4.7 popisuje piatu úroveň ResNet, ktorej prvá konvolučná maska bola upravená na veľkosť 7×7 regiónu záujmu s posunom 1.



Obr. 4.7: Hlavná architektúra siete Faster R-CNN

Pri trénovaní je predpokladané obdobne ako pre Fast R-CNN, že RoI je pozitívny v prípade, že IoU s ground-truth boxom má hodnotu aspoň 0,5. V opačnom prípade je považovaný ako negatívny.

4.2.3 Vytvorenie detektoru pomocou neurónovej siete

Trénovanie neurónovej siete bolo vykonané v už spomínanej cloudovej službe Colab 4.2.1 z dôvodu nedostatku výpočtového výkonu používaného počítača. Služba Colab disponuje predinštalovanými

balíčkami pre vývoj neurónových sietí, pre zvolený typ tréovania je však potrebná inštalácia knižnice Mask-RCNN [2] a zmena verzií používaných balíčkov. Na vytvorenie štruktúry neurónovej siete bola použitá knižnica Keras, ktorá výrazne zjednodušuje vývoj neurónových sietí.

Balíček Mask-RCNN umožňuje segmentáciu obrazu a nájdenie ohraničenia požadovaného objektu, ako aj nájdenie masky, ohraničujúcej objekt. Nakoľko sú prechody pre chodcov jednoduché útvary, ktoré stačí definovať odblžníkovým tvarom, pre potreby nájdenia prechodu pre chodcov na vozovke bude postačujúci ohraničujúci obdĺžnik. Nakoľko podľa [2] je vykonávaná automatická úpravu veľkosti na 1024 * 1024px, na vstup tréovania datasetu je možné zadať obrázky rôznych veľkostí. Pre tréovanie datasetu je použitých 400 obrázkov, z pripravených anotovaných dát, zvyšných 99 bude slúžiť na základné testovanie.

Načítanie informácií o pozícií ohraničujúcich obdĺžnikov je zistené prechádzaním XML súboru nasledujúcim kódom

```
images = root.find('images')
elem = images.find("./*[@file='" + name + "']")
boxes = list()
for box in elem.findall('./box'):
    xmin = int(box.get('left'))
    ymin = int(box.get('top'))
    xmax = int(box.get('left')) + int(box.get('width'))
    ymax = int(box.get('top')) + int(box.get('height'))
    coors = [xmin, ymin, xmax, ymax]
    boxes.append(coors)
```

Výpis 4.2: Algoritmus pre načítanie ohraničujúcich obdĺžnikov z XML súboru

Po nájdení vydefinovaní oblastí, v ktorých sa nechádzajú prechody pre chodcov sú tieto dáta zadane na vstup predtrénovaného modelu. Na tréovanie neurónovej siete je použitý predtrénovaný model COCO. Pre natréovanie modelu bolo potrebných 86 epóch. Po ukončení tréovania bola strata na hodnote 0.1185. Priemerný čas na ukočenie 1 epochy bol 386s čo malo vo výsledku vplyv na celkový čas tréovania ktorý dosiahol takmer hranicu 10 hodín. Po ukončení tréovania bol najlepší model zo služby Google Colab stiahnutý a evaluáciu výsledkov bolo možné vykonávať na osobnom počítači. Implementácia overenia správnosti výsledkov pozostáva z načítania testovacích dát spolu s reálnou pozíciou hľadaného objektu podobne, ako tomu bolo pri tréovaní neurónovej siete. Model vracia predikovanú triedu objektu, ktorá je v prípade nájdenia prechodu pre chodcov vždy rovnaká. Ďalej však bude načrtnuté aj dodatočné využitie Mask-RCNN a jeho možnosť multi-triedovej indentifikácie objektov ako aj použitie skriptu pre prvotné vyfiltrovanie dát pred ich anotáciou pomocou OpenCV.

Implementáciou detektoru, ktorý v obraze dokáže identifikovať prechod pre chodcov za použitia natréovaného modelu je dosiahnutá pomerne vysoká mieru úspešnosti. Pre zlepšenie výsledkov v

testovaní je však možné využiť existujúce riešenie za pomoci knižnice OpenCV a výsledok podmieniť viacerými zdrojmi.

V oblasti spracovania prechodu pre chodcov v rámci jednej krajiny sa javí ako dodatočné určenie prítomnosti prechodu pomocou zvislej dopravnej značky. V prevládajúcej väčšine prípadov je v Slovenskej a Českej Republike pozdĺžne značenie prechodu pre chodcov doplnené aj zvislou dopravnou značkou viditeľnou v obrázkoch 4.8, ktorá je pre vytvorený detektor horšie zameniteľná s inými objektami. Mask-RCNN dokáže detekovať viaceré triedy objektov a preto sa spolu s implementáciou v tejto diplomovej práci javí ako vhodný výber pre definované potreby.



Obr. 4.8: Vertikálne dopravné značky umiestnené pri prechodoch pre chodcov

Táto problematika je priblížená v kapitole 6 v ktorej je natrénovaný základný model s malým počtom tréningových dát na dokázanie možnosti použitia ďalšieho kritéria pre správne vyhodnotenie pozície prechodu.

Kapitola 5

Testovanie

Pre testovanie výsledkov z vytvorených dát boli použité navrhnuté prístupy a niektoré ich kombinácie. Kvôli odlišnosti testovacích dát bol mierne upravený algoritmus ktorý bol využitý pre filtráciu dát určených na vstup neurónovej siete. Dôvodom úpravy je výskyt nových testovacích dát, ktoré pozostávajú zo 100 ručne extrahovaných obrázkov prechodov pre chodcov zo služby Google Street View. Dáta použité na tréning neurónovej siete boli zbierané v úzkych častiach mestských alebo prímestských oblastí. To je zrejme dôvodom, prečo navrhnutý algoritmus nevyhodnocuje výsledky z nových testovacích dát spoľahlivo. Tieto testovacia dáta boli extrahované z rôznych miest v Slovenskej a Českej republike a tak poskytujú dostatočnú variabilitu pre testovanie.

5.1 Použitie modelu natrénovaného neurónovou sieťou

V tejto podkapitole je zdokumentované, ako si implementované riešenia poradia s testovacími dátami. S použitím algoritmu bez tréningu bolo zistené, ako presne dokáže algoritmus zachytiť pozíciu prechodu, pre ďalšie testovanie však bude slúžiť iba ako potvrdzujúci alebo vyvracajúci údaj o zistení výsledku pomocou neurónovej siete.

Z vytvoreného datasetu popísaného v 4.1.2 bolo pre tréning horizontálnych značení prechodov vyčlenených 400 obrázkov. Zvyšných 99 obrázkov bolo použitých pre účely testovania kvality natrénovania siete. Testovanie bolo rozdelené do rôznych kategórií, z ktorých výsledky sú exportované v príslušných tabuľkách a časť výsledkov testovania je pre lepšiu ilustráciu zobrazená v texte práce. Ako prvé bolo vykonané testovanie na určenie prechodu pre chodcov spolu s jeho pozíciou a presnosťou určenia. V tabuľke je zobrazených päť najhoršie vyhodnotených, päť najlepšie vyhodnotených a celkový priemer vyhodnotených testovaných dát 5.1. Všetky percentuálne výsledky sú zaokrúhlené na tri desatinné miesta. Presnosť v nasledujúcich tabuľkách vyjadruje percentuálnu zhodu ohraničujúceho obdĺžniku v anotovaných dátach s reálnym výsledkom vyhodnoteným neurónovou sieťou. Stav je určený skratkami *TP*, *TF*, *FP*, *FN* ktoré budú v testovaní použité pre určenie správnosti odhadu. Skratky vychádzajú z anglických názvov *True/False Negative/Positive* a

kombináciou začiatočných písmen určujú pravdivosť určenia. Viac informácií o použití a význame týchto skratiek je možné nájsť na tomto odkaze ¹.

Tabuľka 5.1: Päť najlepších výsledkov testovania vlastných datasetov

Dadataset	Súbor	Stav	Presnosť (%)
dataset_full_n	crosswalk_141.jpg	TP	92.854
dataset_full_f	crosswalk_370.jpg	TP	91.916
dataset_full_f	crosswalk_371.jpg	TP	91.525
dataset_full_n	crosswalk_149.jpg	TP	91.222
dataset_full_n	crosswalk_151.jpg	TP	90.642

Tabuľka 5.2: Päť najhorších výsledkov testovania vlastných datasetov

Dadataset	Súbor	Stav	Presnosť (%)
dataset_full_n	crosswalk_126.jpg	TP	47.789
dataset_full_f	crosswalk_363.jpg	TP	46.369
dataset_full_f	crosswalk_364.jpg	TP	46.291
dataset_full_n	crosswalk_127.jpg	TP	18.950
dataset_full_n	crosswalk_117.jpg	FN	0

Z tabuliek je možné vidieť, že natrénovaný model má pomerne dobrú úspešnosť v nájdení pozície prechodu. Prechod nenašiel iba v súbore crosswalk_117.jpg, v ktorom bol navyše v atypickej pozícii vzhľadom k vozidlu. V situáciách pri odbočovaní na vedľajšiu cestu je navyše častokrát oveľa menšia šanca zranenia chodca nakoľko vodič býva pri odbočovaní sústredenejší. Zhoda s anotovanými dátami bola vypočítaná ako percentuálny podiel prieniku jednotlivých obrysových obdĺžnikov. Vypočet je možné vidieť v nasledujúcom bloku.

$$overlap = \frac{r3_area}{(r1_area) + (r2_area) - (r3_area)} \quad (5.1)$$

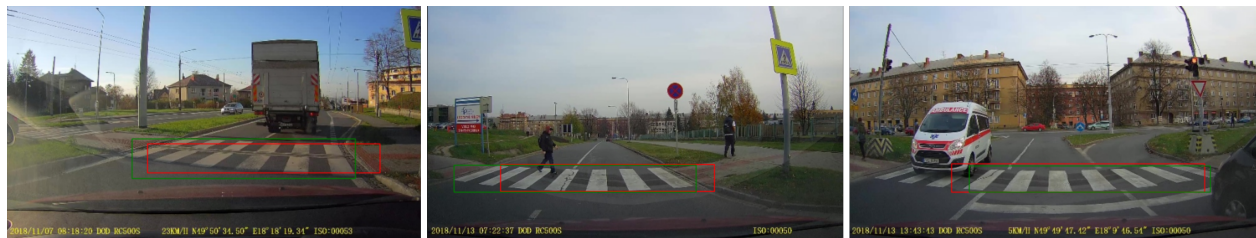
kde $r1_area$ a $r2_area$ sú obsahy obdĺžnikov, ktorých percentuálnu zhodu chceme určiť a

$$r3 = r1 \cap r2 \quad (5.2)$$

$r3_area$ značí obsah prieniku oboch obdĺžnikov.

¹<https://developers.google.com/machine-learning/crash-course/classification/true-false-positive-negative>

Na obrázku 5.1 je možné vidieť porovnanie výsledkov dosiahnutých natrénovaným modelom označených červenou čiarou, k porovnaniu výsledkov anotovaných dát označených zelenou čiarou.



Obr. 5.1: Výsledky testovania datasetu 1. (zelenou čiarou sú označené anotované dáta, červenou výsledok detekcie)

Na obrázku 5.2 je ďalej možné vidieť súbor, v ktorom sa prechod nepodarilo nájsť a ďalšie porovnanie výsledkov z datasetu dataset_n.



Obr. 5.2: Výsledky testovania datasetu 2. (zelenou čiarou sú označené anotované dáta, červenou výsledok detekcie)

Tabuľka 5.3: Celkový priemer výsledkov testovacích dát

Názov	Prechod nájdený v)	Priemerná zhoda s anot. dátami - Presnosť (%)
Celkovo	97/98 obrázkov (99.329)	71.832

5.2 Použitie datasetov extrahovaných zo služieb tretích strán

Datasety extrahované z dostupných zdrojov svoju úlohu v rámci testovania neurónovej siete splnili, avšak výsledky boli pomerne dobré najmä kvôli nízkej diverzite prostredí a typov prechodov. Práve z tohoto dôvodu boli z voľne dostupných zdrojov (Google Street View a mapy.cz Panoráma) ručne získané obrázky prechodov. Obrázky boli extrahované z väčších miest naprieč celým Slovenskom a Českou Republikou. Zozbierané dáta z rôznych miest poskytujú väčšiu variabilitu a vhodnejší spôsob otestovania spoľahlivosti neurónovej siete. Navyše sa v dátach nachádzajú aj vertikálne dopravné

značenia prechodov, čo otvára nové možnosti overenia. Zo spomínaných zdrojov bolo zozbieraných 100 obrázkov prechodov pre chodcov. Tieto dáta však neboli anotované, preto v súvislosti s ich použitím bude rozlišované, či algoritmus prechod v obraze dokázal detekovať alebo nie. Výsledky o presnosti detekcie prechodu budú zozbierané ručne. Na brázku 5.3 je možné vidieť niekoľko prechodov extrahovaných v rámci tohoto experimentu. Na obrázku je zrejماً rozličnosť prostredí pre komplexnejšie testovanie.



Obr. 5.3: Nový testovací dataset vytvorený pomocou služieb tretích strán

Testovanie na novom datasete spočíva z oddeleného behu algoritmov, pomocou ktorých je vyhodnocované, či sa prechod v obraze nachádza a v prípade neurónovej siete bude zistená aj pozícia prechodu pre chodcov. Ako prvý bude otestovaný algoritmus bez trénovania. Výsledkom tohoto testu je číslo na stupnici od 1-4 ktoré značí, s akou pravdepodobnosťou sa v obraze prechod nachádza. Algoritmus sa rozhoduje na základe viacerých faktorov akým je napr. počet vyfiltrovaných čiar prechodu. V prípade potvrdenia funkčnosti algoritmu môže slúžiť ako rozhodovacia váha v prípade zlého vyhodnotenia neurónovou sieťou.

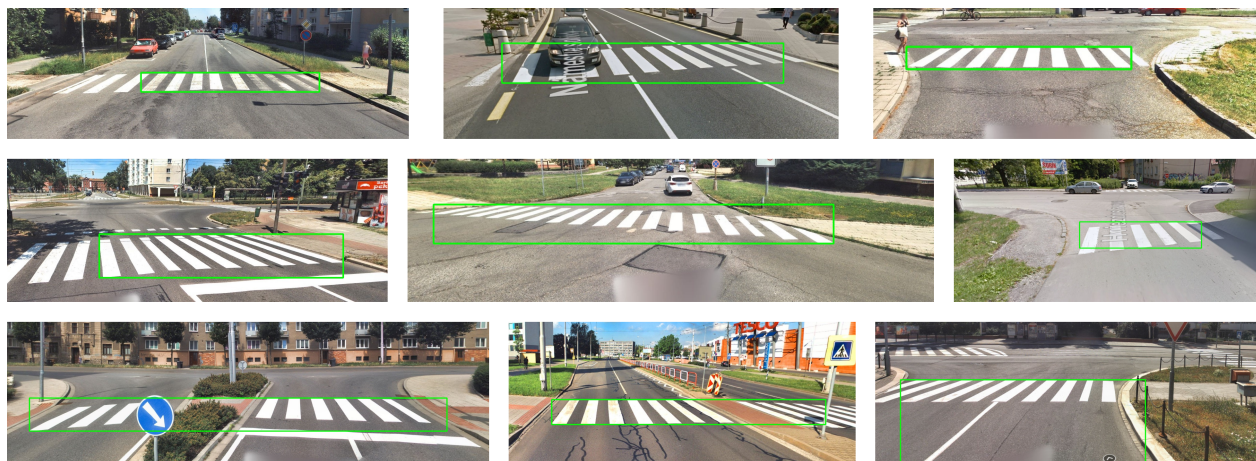
Výsledky testovania sú zobrazené v nasledujúcej tabuľke 5.4. Posudzovanie jednotlivých výsledkov prebehlo manuálne. Nakoľko dáta neboli anotované, automatický skript vyexportoval všetky testovacie obrázky s označenými prechodmi a istotou, s akou si algoritmus myslí, že sa v obraze prechod nachádza. Za pozitívne sa považujú výsledky, kedy skript označil prechod na mieste, na ktorom sa skutočne nachádza, alebo neoznačil žiadny prechod na obrázku, na ktorom sa žiaden prechod nenachádza.

Algoritmus prekvapivo spoľahlivo odhaľuje prechody s nízkou mierou amortizácie, ktoré sú kvalitne nakreslené a pri ktorých fotografie majú dobrú kvalitu a vhodné počasie. Pri hodnote rozhodnutia s číslom 3, kedy si je algoritmus takmer úplne istý, že sa na danom mieste prechod nachádza je

Tabuľka 5.4: Testovanie pomocou algoritmu bez trénovania na datasete zo služieb tretích strán

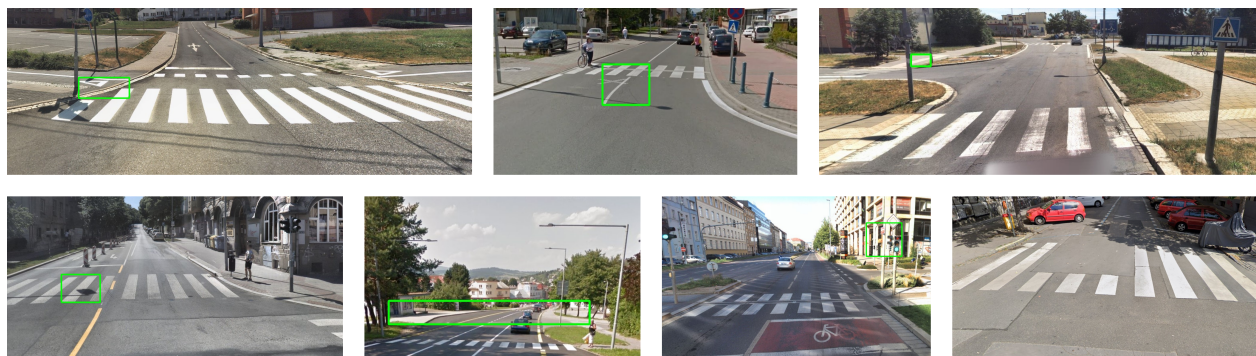
Rozhodnutie (0-3)	Počet výsledkov	Z toho pozitívnych	Z toho negatívnych
0	19	3	16
1	23	14	9
2	12	8	4
3	48	47	1
Celkovo	102	72	30

falošný negatívny iba 1 výsledok. Pri ostatných výsledkoch dokonca pomerne presne označil pozíciu prechodu (viď obrázok 5.4)



Obr. 5.4: Správne vyhodnotené výsledky použitím neurónovej siete

Zobrazenie niektorých nesprávne vyhodnotených výsledkov testu je dostupné na obrázku 5.5



Obr. 5.5: Nesprávne vyhodnotené výsledky použitím neurónovej siete

Ďalším krokom k overeniu vhodnosti použitia algoritmov je otestovanie tohoto datasetu na natrénovanom modeli neurónovej siete Mask-RCNN. Nakoľko dáta neboli anotované, výsledky budú zhodnotené podobne ako pri využití algoritmu bez tréningu. Miera zhody medzi týmito prístupmi bude zhodnotená v ďalších testoch.

Použitie natrénovaného modelu malo veľký vplyv na kvalitu výsledkov. Natrénovaný model pomerne dobre odhalil aj slabšie viditeľné alebo neúplné prechody (viď obrázok 5.6), nakoľko v tréningových dátach boli prechody rôznej miery amortizácie. Slabší kontrast medzi čiarami a vozovkou mal však za následok aj niekoľko falošných pozitívnych výsledkov.



Obr. 5.6: Zle viditeľné alebo neúplné prechody a ich vyhodnotenie natrénovaným modelom

Na obrázku 5.7 je možné vidieť správne vyhodnotené výsledky spolu s tromi falošnými pozitívnymi výsledkami, ktoré sú ale stále súčasťou pozície prechodu.



Obr. 5.7: Správne alebo čiastočne správne vyhodnotené výsledky natrénovaným modelom

Niekoľko falošných negatívnych výsledkov je dostupných na obrázku 5.8 a komplexný prehľad o vykonanom testovaní je možné vidieť v nasledujúcom zozname. Test pozostával zo 102 súborov, z ktorých po vyhodnotení natrénovaným modelom vyšli nasledovné štatistiky:

- 10 obrázkov boli nesprávne výsledky ktoré budú rozlíšené v testovaní oboch prístupov v ďalšej podsekcii

- 19 obrázkov bolo vyhodnotených správne, avšak model v obraze našiel aj ďalší, neexistujúci prechod (ten častokrát zasahoval do pôvodného prechodu)
- 73 obrázkov bolo vyhodnotených správne aj s umiestnením prechodu v obraze



Obr. 5.8: FN výsledky pomocou natrénovaného modelu

Ďalším testovaním je možné potvrdzovať správnosť jednotlivých výsledkov posúdením oboch metód na jednom obraze súčasne.

5.3 Použitie spojenia neurónovej siete a prístupu bez tréovania

V tejto sekcii bude porovnané spojenie oboch navrhnutých prístupov a ich výsledky vzhľadom na presnosť určenia prechodu pre chodcov. V testovacích dátach je zahrnutý dataset vytvorený zo služieb tretích strán. V tabuľke 5.5 je možné vidieť rozloženie správnosti určenia prechodov v obrazoch s oboma prístupmi.

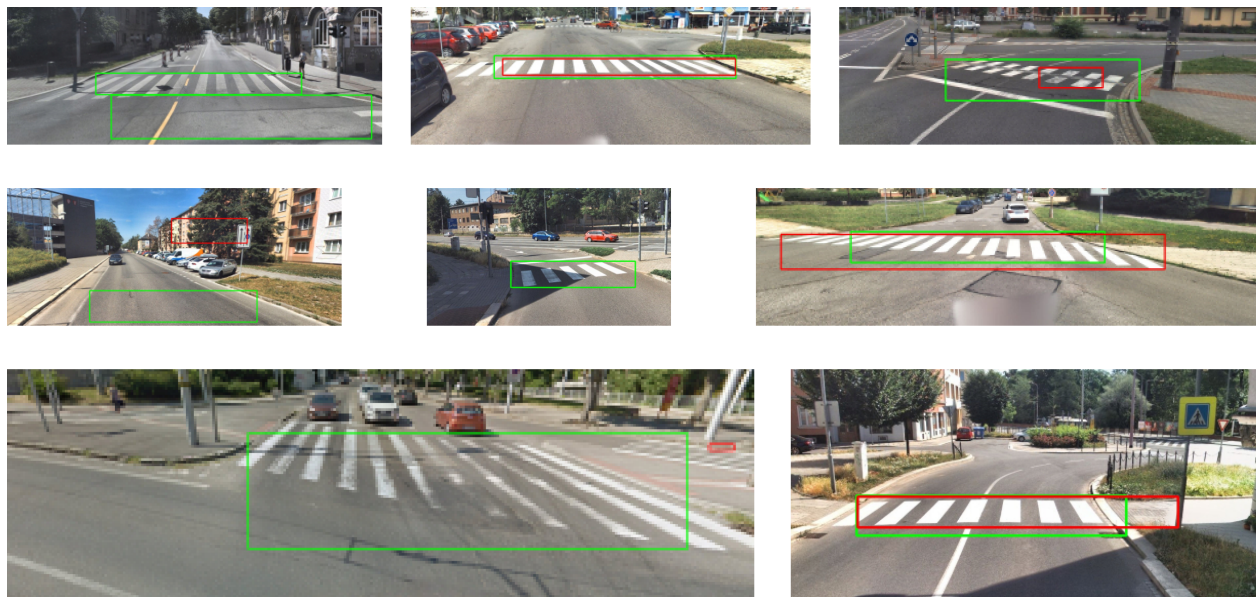
Tabuľka 5.5: Rozloženie správnosti testovacích dát pomocou oboch prístupov

Použitý algoritmus	TP	FP	FN
Bez tréovania	62	19	21
S tréovaním (Neurónová sieť)	74	4	7

Nakoľko bolo zistené, že prístup s tréovaním v príslušnom datasete určil približne 90% dát správne a prístup bez tréovania mal razantne horšie výsledky, určenie konečného výsledku je vykonané nasledovne:

- TP výsledky prístupu s tréovaním sú predpokladané ako správne určené, prístup bez tréovania môže potvrdiť daný výsledok. V prípade vyvrátenia výsledku prístupom bez tréovania sa v obraze naďalej bude predpokladať prítomnosť prechodu kvôli menšej presnosti tohoto prístupu
- TF výsledky prístupu s tréovaním môžu byť vyvrátené v prípade TP prístupu bez tréovania
- Ostatné falošné výsledky v oboch prístupoch sa budú považovať za neprítomnosť prechodu

Z tohoto určenia vyplýva, že algoritmus predpokladá prítomnosť prechodu v ďalších šiestich prípadoch. To znamená, že presnosť určenia sa v porovnaní s použitím prístupu s trénovaním zvýšila o 6,12%. Presnosť určenia prechodu pre chodcov je však stále zaťažená viacerými chybnými FP určeními prístupu s trénovaním. Zlúčenie oboch prístupov preto dokáže s väčšou istotou povedať, či sa v obraze prechod nachádza alebo nie, nedokáže však spoľahlivo určiť presnú pozíciu prechodu. Na obrázku 5.9 je možné vidieť niektoré z výsledkov tohoto testovania.



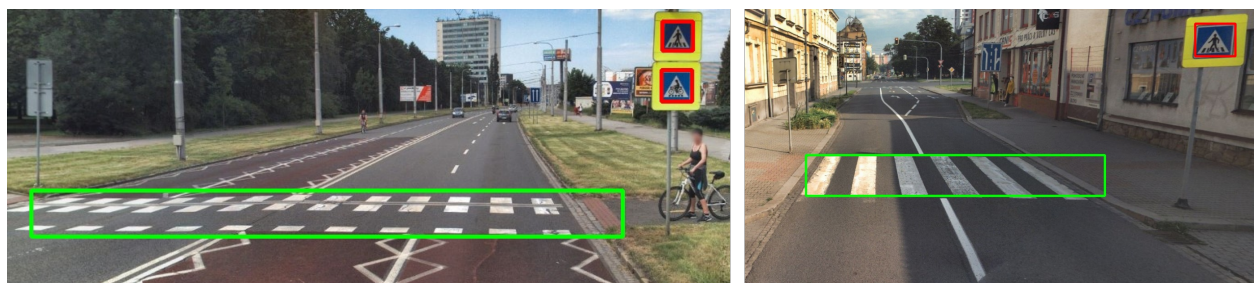
Obr. 5.9: Zobrazenie niektorých výsledkov testovania oboch prístupov (zelenou čiarou sú výsledky prístupu s trénovaním, červenou sú výsledky algoritmu bez trénovania)

Kapitola 6

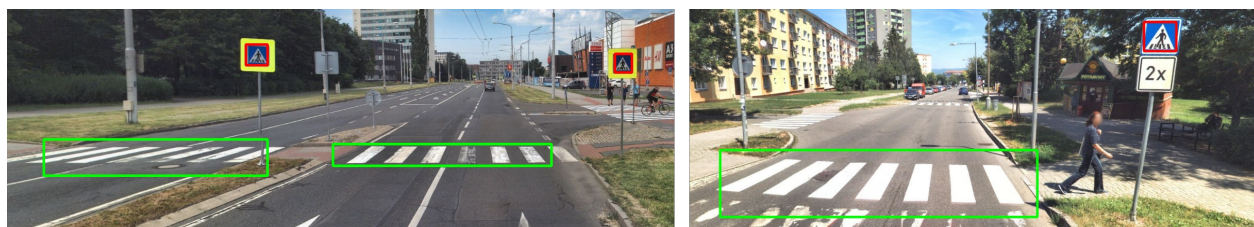
Možnosti zlepšenia

Natrénovaný model dátami z vytvorených datasetov sa pri testovaní javí ako dostatočný pre identifikáciu prechodov v testovacích dátach extrahovaných zo služieb tretích strán. Model však nebol testovaný v reálnej záťaži a s dátami, ktoré sa môžu nápadne podobáť prechodu pre chodcov. Pri využití prístupu bez tréningu je taktiež dôležité vymedzenie správneho regiónu záujmu, čo by pri statickom umiestnení kamery v automobile malo byť jednoducho realizovateľné. Stále je však obrovský priestor na zlepšenie najmä použitím prístupu bez tréningu, aj keď v tejto diplomovej práci bol použitý z väčšej časti iba na extrakciu dát pre tvorbu datasetu. Nakoľko prístup s tréningom má v tejto diplomovej práci väčšinové zastúpenie, bol v rámci tréningu modelu vykonaný pokus o doplnenie tréningových dát o vertikálne značenie prechodov, ktoré sa v niektorých z obrázkov periodicky opakuje. V Slovenskej a Českej republike má táto značka takmer identický vzhľad, preto sa javí ako vhodná možnosť na utvrdenie algoritmu pri určovaní, či sa v obraze prechod pre chodcov naozaj nachádza. Z tohoto dôvodu bola vykonaná anotácia dát vertikálnych značiek prechodov pre chodcov a bol natrénovaný nový model s malým tréningovým datasetom. Nakoľko použitá neurónová sieť je určená práve na multi-triedové učovanie objektov v obraze, pôvodný tréningový algoritmus je potrebné doplniť o novú tréningovú triedu s príslušnými anotovanými dátami.

Testovanie tohoto prístupu však ostáva otvorené, nakoľko tento prístup nemá dostatočne veľký dataset. Pre overenie funkčnosti a použiteľnosti dodatočného overenia prítomnosti vertikálnej dopravnej značky prechodu bol natrénovaný model použitý aspoň na malé množstvo testovacích dát. Výsledky tohto návrhu na zlepšenie je možné vidieť na obrázkoch 6.1 a 6.2.



Obr. 6.1: Zobrazenie dát vyhodnotených modelom natrénovaným pomocou vertikálnych dopravných značiek prechodov (zelenou farbou model označuje pozdĺžne značenie, červenou vertikálne)



Obr. 6.2: Zobrazenie dát vyhodnotených modelom natrénovaným pomocou vertikálnych dopravných značiek prechodov 2 (zelenou farbou model označuje pozdĺžne značenie, červenou vertikálne)

Ako ďalší postup by sa predpokladal návrh algoritmu, ktorý by spolu s prístupom bez tréovania a s použitím nového natrénovaného modelu najskôr určil, či sa v obraze prechod nachádza. Pokiaľ by boli potvrdené aspoň dve z troch kritérií, je pomerne veľká pravdepodobnosť že sa prechod v obraze naozaj nachádza. Kritéria by z rozboru tejto diplomovej práce mohli byť určené na základe výsledku bez tréovania, nájdenia pozície horizontálneho značenia a nájdenia vertikálneho značenia prechodu pre chodcov.

Ďalej sa naskytuje dodatočné overenie výsledkov z výstupu neurónovej siete prístupom bez tréovania s prísnejším nastavením parametrov. V predpokladanej pozícii prechodu pre chodcov a s dostatočne dobrým vymedzením regiónu záujmu by mohli byť v prístupe bez tréovania nastavené prísnejšie pravidlá pre nájdenie prechodu, čo by mohlo mať za následok odfiltrovanie FP výsledkov na úkor zvýšenia aj falošných negatívnych výsledkov, ktoré by však mala spoľahlivo pokryť natrénovaná neurónová sieť.

Pri vytváraní komplexného obrazu o udalostiach v blízkosti vozidla by mohlo hrať zaujímavú úlohu aj rozpoznávanie iných objektov ako priamo značenie prechodov. Úplne autonómne vozidlá by pri spracovaní celkového obrazu o vozovke mohli brať v úvahu napr. chodcov stojacich alebo pohybujúcich sa blízko pri vozovke. Predikcia správania sa chodcov môže viesť k ďalším výstražným signálom, ktoré v kombinácii s detekciou prechodu pre chodcov môžu mať razantný vplyv na potencionálne nebezpečenstvo.

Kapitola 7

Záver

Témou tejto diplomovej práce bola analýza prechodov pre chodcov pomocou obrazov. V úvode a v kapitolách 1 a 2 bol čitateľ oboznámený o problematike prechodov pre chodcov a ich rôznych typoch. Najvýznamnejšie využitie analýzy prechodov bolo vyhodnotené v sektore autonómnych vozidiel a v oblasti zrakového postihnutia chodcov. V oboch prípadoch môže mať návrh detektoru a jeho spoľahlivosť značný vplyv na bezpečnosť cestnej premávky a chodcov. Analýza dostupných riešení v kapitole 3 ukázala vhodných kandidátov pre spracovanie obrazu a rôzne prístupy k jeho analýze. Pre použitie v tejto diplomovej práci bol vybraný ako prístup bez tréningovania, tak použitie nerónových sietí. Prístup bez tréningovania bol navrhnutý a implementovaný pomocou knižnice OpenCV a je bližšie popísaný v kapitole 4.1. Pre návrh detektoru za pomoci neurónovej siete bolo potrebné zvoliť typ neurónovej siete a získať potrebné dáta pre jej natréningovanie. Tie boli získané pomocou navrhnutého detektoru s použitím OpenCV. Následne boli získané dáta ručne vyfiltrované a anotované pomocou webovej služby Imglab čo je bližšie popísané v kapitole 4.1.2. Kvôli nedostatku potrebného výpočtového výkonu autorovho počítača bol model natréningovaný pomocou služby Google Colab. Neurónová sieť M-RCNN pre natréningovanie modelu bola vybraná kvôli schopnosti určenia polohy objektu a možnosti určenia viacerých tried objektov. Po natréningovaní modelu, ktoré je popísané v kapitole 4.2 bolo vykonané testovanie na testovacích dátach ďalej zdokumentované v kapitole 5. Nakoľko boli testovacie dáta extrahované z datasetu vytvoreného v podobných podmienkach, kvôli presnejšiemu otestovaniu bol zo služieb tretích strán ako sú Google Street View a Mapy.cz Panoráma vytvorený nový testovací dataset s dátami z rôznych oblastí Slovenskej a Českej republiky a v rôznych podmienkach podnebia. Rôzne druhy testov boli vykonané v oboch datasetoch a výsledky boli zdokumentované v texte diplomovej práce.

Dodatočne bol z existujúcich dát extrahovaný nový dataset označujúci vertikálne značenie prechodov pre chodcov a pomocou použitej neurónovej siete bol natréningovaný nový multi-triedový model. Toto doporučenie nebolo bližšie testované, je však použiteľné a výsledky sú zobrazené v návrhoch na zlepšenie 6. S ostatnými asistenčnými systémami vo vozidle má detekcia prechodov tvoriť prevenciu pred kolíziami na cestách a poskytovať dodatočné informácie vodičovi vozidla.

Literatúra

- [1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283, 2016.
- [2] Waleed Abdulla. Mask r-cnn for object detection and instance segmentation on keras and tensorflow. https://github.com/matterport/Mask_RCNN, 2017.
- [3] Dana H Ballard. Generalizing the hough transform to detect arbitrary shapes. In *Readings in computer vision*, pages 714–725. Elsevier, 1987.
- [4] Jason David Banich. Zebra crosswalk detection assisted by neural networks. 2016.
- [5] Michele Bertoncello and Dominik Wee. Ten ways autonomous driving could redefine the automotive world. *McKinsey & Company*, 6, 2015.
- [6] Gary Bradski and Adrian Kaehler. Opencv. *Dr. Dobb's journal of software tools*, 3, 2000.
- [7] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. Ö'Reilly Media, Inc.", 2008.
- [8] Tiago Carneiro, Raul Victor Medeiros Da Nóbrega, Thiago Nepomuceno, Gui-Bin Bian, Victor Hugo C De Albuquerque, and Pedro Pedrosa Reboucas Filho. Performance analysis of google colabory as a tool for accelerating deep learning applications. *IEEE Access*, 6:61677–61685, 2018.
- [9] Ruiqi Cheng, Kaiwei Wang, Kailun Yang, Ningbo Long, Weijian Hu, Hao Chen, Jian Bai, and Dong Liu. Crosswalk navigation for people with visual impairments on a wearable device. *Journal of Electronic Imaging*, 26(5):053025, 2017.
- [10] SAE On-Road Automated Vehicle Standards Committee et al. Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems. *SAE Standard J*, 3016:1–16, 2014.

- [11] James Coughlan and Huiying Shen. A fast algorithm for finding crosswalks using figure-ground segmentation. In *2nd Workshop on Applications of Computer Vision, in conjunction with ECCV*, volume 5, 2006.
- [12] Donald Davidson. Actions, reasons, and causes. *The journal of philosophy*, 60(23):685–700, 1963.
- [13] Brian Lystgaard Due. The future of smart glasses: An essay about challenges and possibilities with smart glasses. *Working papers on interaction and communication*, 1(2):1–21, 2014.
- [14] Philippe Foucher, Yazid Sebsadji, Jean-Philippe Tarel, Pierre Charbonnier, and Philippe Nicolle. Detection and recognition of urban road markings using images. In *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 1747–1752. IEEE, 2011.
- [15] Andrea Frome, German Cheung, Ahmad Abdulkader, Marco Zennaro, Bo Wu, Alessandro Bissacco, Hartwig Adam, Hartmut Neven, and Luc Vincent. Large-scale privacy protection in google street view. In *2009 IEEE 12th international conference on computer vision*, pages 2373–2380. IEEE, 2009.
- [16] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.
- [17] Antonio Gulli and Sujit Pal. *Deep learning with Keras*. Packt Publishing Ltd, 2017.
- [18] Anselm Haselhoff and Anton Kummert. On visual crosswalk detection for driver assistance systems. In *2010 IEEE Intelligent Vehicles Symposium*, pages 883–888. IEEE, 2010.
- [19] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.
- [20] Danilo Cáceres Hernández, Alexander Filonenko, Dongwook Seo, and Kang-Hyun Jo. Crosswalk detection based on laser scanning from moving vehicle. In *2015 IEEE 13th International Conference on Industrial Informatics (INDIN)*, pages 1515–1519. IEEE, 2015.
- [21] Danilo Caceres Hernandez, Van-Dung Hoang, and Kang-Hyun Jo. Lane surface identification based on reflectance using laser range finder. In *2014 IEEE/SICE International Symposium on System Integration*, pages 621–625. IEEE, 2014.
- [22] Volodymyr Ivanchenko, James Coughlan, and Huiying Shen. Detecting and locating crosswalks using a camera phone. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–8. IEEE, 2008.

- [23] Christoph G Keller and Darius M Gavrilă. Will the pedestrian cross? a study on pedestrian path prediction. *IEEE Transactions on Intelligent Transportation Systems*, 15(2):494–506, 2013.
- [24] Reinhard Klette and Auckland CITR. Vision-based driver assistance systems. *Available: researchgate.net/publication/272199860 Visionbased Driver Assistance Systems*, 2015.
- [25] Sebastian Koehler, Michael Goldhammer, Sebastian Bauer, Stephan Zecha, Konrad Doll, Ulrich Brunsmann, and Klaus Dietmayer. Stationary detection of the pedestrian? s intention at intersections. *IEEE Intelligent Transportation Systems Magazine*, 5(4):87–99, 2013.
- [26] Samuel Kurath, Raphael Das Gupta, Stefan Keller, et al. Osmdeepod-object detection on orthophotos with and for vgi. In *GI Forum*, volume 2, pages 173–188, 2017.
- [27] Robert Laganière. *OpenCV Computer Vision Application Programming Cookbook Second Edition*. Packt Publishing Ltd, 2014.
- [28] Ludwig Lausser, Friedhelm Schwenker, and Günther Palm. Detecting zebra crossings utilizing adaboost. In *ESANN*, pages 535–540, 2008.
- [29] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.
- [30] IHS Markit. Autonomous vehicle sales forecast to reach 21 mil, globally in 2035, according to ihs automotive, 2016.
- [31] Andrew Ng, Jiquan Ngiam, Chuan Yu Foo, Yifan Mai, and Caroline Suen. Ufddl tutorial, 2013. URL http://ufddl.stanford.edu/wiki/index.php/UFLDL_Tutorial, 2015.
- [32] Anantha Pillai et al. Virtual reality based study to analyse pedestrian attitude towards autonomous vehicles. 2017.
- [33] Amir Rasouli and John K Tsotsos. Autonomous vehicles that interact with pedestrians: A survey of theory and practice. *IEEE Transactions on Intelligent Transportation Systems*, 2019.
- [34] Kresimir Romic, Irena Galic, Hrvoje Leventic, and Kresimir Nenadic. Real-time multiresolution crosswalk detection with walk light recognition for the blind. *Advances in Electrical and Computer Engineering*, 18(1):11–21, 2018.
- [35] Sarah Schmidt and Berthold Faerber. Pedestrians at the kerb—recognising the action intentions of humans. *Transportation research part F: traffic psychology and behaviour*, 12(4):300–310, 2009.

- [36] Friederike Schneemann and Patrick Heinemann. Context-based detection of pedestrian crossing intention for autonomous driving in urban environments. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2243–2248. IEEE, 2016.
- [37] Stephen Se. Zebra-crossing detection for the partially sighted. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, volume 2, pages 211–217. IEEE, 2000.
- [38] Stephen Se and Michael Brady. Road feature detection and estimation. *Machine Vision and Applications*, 14(3):157–165, 2003.
- [39] Tadayoshi Shioyama, Haiyuan Wu, Yuna Nishibe, Naoki Nakamura, and Suguru Kitawaki. Image analysis of crosswalk. In *Proceedings 11th International Conference on Image Analysis and Processing*, pages 168–173. IEEE, 2001.
- [40] B Soheilian, N Paparoditis, D Boldo, and JP Rudant. 3d zebra-crossing reconstruction from stereo rig images of a ground-based mobile. 2006.
- [41] Vedat Tümen and Burhan Ergen. Intersections and crosswalk detection using deep learning and image processing techniques. *Physica A: Statistical Mechanics and its Applications*, 543:123510, 2020.
- [42] Paul Viola, Michael Jones, et al. Robust real-time object detection. *International journal of computer vision*, 4(34-47):4, 2001.
- [43] Greg Welch, Gary Bishop, et al. An introduction to the kalman filter. 1995.
- [44] Jonathan S Yedidia, William T Freeman, and Yair Weiss. Generalized belief propagation. In *Advances in neural information processing systems*, pages 689–695, 2001.

Dodatok A

Zoznam priložených súborov a zložiek a popis testovania kódu

V priloženom archíve sa nachádza nasledujúci zoznam zložiek a súborov, ktoré priamo súvisia s otestovaním vytvorených algoritmov použitých v tejto diplomovej práci:

```
/
├── data
│   ├── test ... Zložka obsahuje predpripravené súbory na
│   │           testovanie pomocou súboru main.py.
│   └── output_test ... Zložka obsahuje ukážku testovaných dát
│   │                   vyhodnotených zo zložky test.
├── colab_train.py ... Súbor obsahuje zdrojový kód použitý na
│   │               natrénovanie neurónovej siete v službe Google
│   │               Colab.
├── main.py ... V tomto súbore sa spúšťa hlavný program ktorý
│   │           vyhodnocuje súbory zo zložky test.
├── opencv_detector.py ... Súbor obsahuje skript pre detekciu prechodov
│   │                   pomocou OpenCV.
├── mask_finder.py ... Súbor obsahuje pomocný skript pre nájdenie
│   │               prahovania pre zvolený vstup.
├── mask_rcnn_detector.py ... Súbor obsahuje skript pre detekciu prechodov
│   │                   pomocou Mask RCNN.
└── model.h5 ... Súbor obsahuje natrénovaný model pre detekciu
    │           vertikálnych a horizontálnych značiek prechodov
    │           pre chodcov.
```

Pre použitie priložených súborov je potrebná inštalácia špecifických verzií balíčkov. Pre testovanie bolo použité virtuálne prostredie *Python 3.6* a potrebné balíčky boli inštalované vo vývojovom prostredí PyCharm, ktoré je zdarma pre edukačné účely. Okrem bežných balíčkov, ktoré sú automaticky predinštalované a pripravené na použitie je potrebné stiahnuť a nainštalovať balíčky *Keras 2.1.0*, *mask-rcnn 2.1*, *opencv-python 4.5.1.48*, *tensorflow 1.15.3* a *matplotlib 3.3.4*. Funkčnosť bola overená na uvedených verziách balíčkov, ktoré môžu pri inštalácii doinštalovať pridružené balíčky. Verzie pridružených balíčkov nebudú bližšie špecifikované, nakoľko by vývojové prostredie malo byť schopné balíčky nainštalovať automaticky.

Pre otestovanie vlastných obrázkov obsahujúcich prechod pre chodcov je potrebné vložiť súbory do zložky *data/test* a spustiť skript *main.py*. Výsledky s naznačenými detekciami prechodov sú ukladané do zložky *data/output_test*. Červenou farbou je do výstupného súboru zaznačený výsledok pomocou algoritmu bez tréningu a zelenou farbou je naznačený výsledok modelu natrénovaného pomocou neurónovej siete.

Adresár súborov a súbory samotné obsahujú rôzne pomocné funkcie a zlomky kódu, ktoré viedli alebo súvisia s filtráciou anotovaných dát a detekciou prechodov pre chodcov.

V adresárovej štruktúre sa nachádza aj natrénovaný model neurónovej siete *model.h5*, ktorý je schopný detekovať vertikálne a horizontálne značenie prechodov pre chodcov.